

010027709 \*\*Image available\*\*

WPI Acc No: 1994-295422/199437

XRPX Acc No: N94-232478

Image processing system performing image compression and expansion - has control unit with separate processors for total control and for variable length conversion and quantisation of data

Patent Assignee: MITSUBISHI DENKI KK (MITQ ); MITSUBISHI ELECTRIC CORP (MITQ )

Inventor: HANAMI A; ISHIHARA K; KUMAKI S; MATSUMURA T; NAKAGAWA S; SEGAWA H ; ISCHIHARA K

Number of Countries: 003 Number of Patents: 013

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
DE 4408522	A1	19940922	DE 4408522	A	19940314	199437 B
FR 2702862	A1	19940923	FR 943206	A	19940318	199438
FR 2707779	A1	19950120	FR 948221	A	19940704	199509
FR 2707780	A1	19950120	FR 943206	A	19940318	199509
			FR 948222	A	19940704	
JP 7240844	A	19950912	JP 9420937	A	19940218	199545
DE 4447555	A1	19951130	DE 4408522	A	19940314	199602
			DE 4447555	A	19940314	
DE 4447554	A1	19951207	DE 4408522	A	19940314	199603
			DE 4447554	A	19940314	
DE 4447552	A1	19951214	DE 4408522	A	19940314	199604
			DE 4447552	A	19940314	
DE 4447553	A1	19951214	DE 4408522	A	19940314	199604
			DE 4447553	A	19940314	
DE 4408522	C2	19961205	DE 4408522	A	19940314	199702
DE 4447552	C2	19990311	DE 4408522	A	19940314	199914
			DE 4447552	A	19940314	
DE 4447554	C2	19990819	DE 4408522	A	19940314	199937
			DE 4447554	A	19940314	
DE 4447553	C2	19990819	DE 4408522	A	19940314	199937
			DE 4447553	A	19940314	

Priority Applications (No Type Date): JP 9420937 A 19940218; JP 9360335 A 19930319; JP 93301716 A 19931201

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
DE 4408522	A1	134		G06F-015/66	
FR 2707780	A1			H04N-007/24	Div ex application FR 943206
JP 7240844	A	74		H04N-001/41	
DE 4447555	A1	1		G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
DE 4447554	A1	1		G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
DE 4447552	A1	1		G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
DE 4447553	A1	1		G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
DE 4408522	C2	129		G06T-009/00	Div in patent DE 4447552
					Div in patent DE 4447553
					Div in patent DE 4447554
					Div in patent DE 4447555
DE 4447552	C2			G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
DE 4447554	C2			G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
DE 4447553	C2			G06T-009/00	Div ex application DE 4408522
					Div ex patent DE 4408522
FR 2707779	A1			H04N-007/24	

Abstract (Basic): DE 4408522 A

The image processing system has a host computer coupled to a control unit that has a processor for overall control (11) that operates in a pipeline mode for the direct cosine conversion,

quantisation and processing of data of variable length. The system also includes a processor (12) for variable length code conversion.

Compressed pixel data is held in memory and is accessed for conversion purposes. The use of separate units improves the speed of executor.

USE/ADVANTAGE - Higher speed image processing.

D20

Dwg.9/95

Title Terms: IMAGE; PROCESS; SYSTEM; PERFORMANCE; IMAGE; COMPRESS; EXPAND;  
CONTROL; UNIT; SEPARATE; PROCESSOR; TOTAL; CONTROL; VARIABLE; LENGTH;  
CONVERT; QUANTUM; DATA

Derwent Class: T01; W04

International Patent Class (Main): G06F-015/66; G06T-009/00; H04N-001/41;  
H04N-007/24

International Patent Class (Additional): G06T-001/20; H03M-007/30;  
H04N-007/13; H04N-007/26; H04N-007/32

File Segment: EPI

Manual Codes (EPI/S-X): T01-D02; T01-J10A1; T01-J10B; W04-P01A; W04-P01A3



①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ Offenlegungsschrift  
⑩ DE 44 08 522 A 1

⑤1 Int. Cl.<sup>5</sup>:  
G 06 F 15/66  
H 04 N 7/13

②1 Aktenzeichen: P 44 08 522.2  
②2 Anmeldetag: 14. 3. 94  
④3 Offenlegungstag: 22. 9. 94

DE 44 08 522 A 1

③0 Unionspriorität: ③2 ③3 ③1  
19.03.93 JP 5-60335 01.12.93 JP 5-301716  
18.02.94 JP 6-20937

⑦1 Anmelder:  
Mitsubishi Denki K.K., Tokio/Tokyo, JP

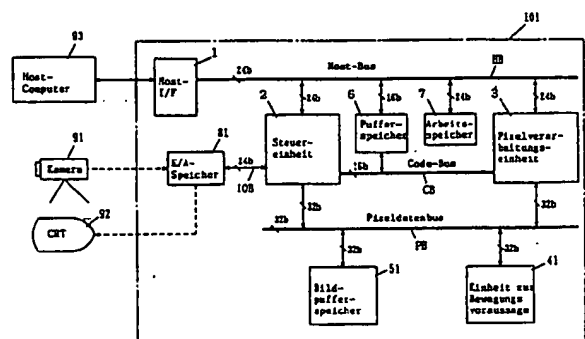
⑦4 Vertreter:  
Prüfer, L., Dipl.-Phys.; Materne, J., Dipl.-Phys.  
Dr.rer.nat.habil., Pat.-Anwälte, 81545 München

⑦2 Erfinder:  
Matsumura, Tetsuya, Itami, Hyogo, JP; Nakagawa,  
Shinichi, Itami, Hyogo, JP; Segawa, Hiroshi, Itami,  
Hyogo, JP; Ishihara, Kazuya, Itami, Hyogo, JP;  
Kumaki, Satoshi, Itami, Hyogo, JP; Hanami, Atsuo,  
Itami, Hyogo, JP

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤4 Vorrichtung und Verfahren zur Bilddatenverarbeitung, die zur Verarbeitung von Bilddaten mit hoher Geschwindigkeit in der Lage sind

⑤7 Eine Vorrichtung zur Bildkompression/-expansion gemäß einer internationalen Standardspezifikation zur Bildkompression umfaßt: eine Steuereinheit (2) mit einer Gesamtsteuerverarbeitung und einem Prozessor für variable Längen, eine Pixelverarbeitungseinheit (3) zur diskreten Kosinsumwandlung und zur Quantisierung, einen SDRAMs verwendenden Bildpufferspeicher (51), eine Einheit zur Bewegungsvoraussage (41), einen Pixeldatenbus (PB) und einen Codebus (CB). Der in der Steuereinheit (2) vorgesehene Gesamtsteuerungsprozessor bewirkt eine Pipelinesteuerung zur Bildkompression/-expansion. Als ein Ergebnis kann die Bildkompression/-expansion durch unabhängig voneinander betriebene Betriebseinheiten ausgeführt werden, und eine Pipelineverarbeitung wird bewirkt. Daher kann eine zum Betrieb bei höherer Geschwindigkeit geeignete Vorrichtung zur Bildkompression/-expansion vorgesehen werden.



DE 44 08 522 A 1

## Beschreibung

Die vorliegende Erfindung betrifft im allgemeinen eine Vorrichtung zur Bilddatenverarbeitung und ein Verfahren zur Bildverarbeitung. Insbesondere betrifft sie eine Vorrichtung und eine Methode zur Bilddatenverarbeitung, durch welche eine Kompression und/oder Expansion der Bilddaten effektiv mit hoher Geschwindigkeit ausgeführt werden kann.

Internationale Standardspezifikationen zur Bilddatenkompression und -expansion sind durch die "International Organization for Standardization", nachstehend als "ISO" bezeichnet, das "International Telegraph and Telephone Consultative Committee", nachstehend als "CCITT" bezeichnet, derzeit als ITU-T bekannt, und das "International Electrical Committee", nachstehend als "IEC" bezeichnet, aufgestellt worden. Bei den internationalen Standardspezifikationen ist die JPEG-Spezifikation durch die "Joint Photographic Expert Group" der ISO und des CCITT aufgestellt worden, und sie legt Kompressions- und Expansionsalgorithmen für Farb-Festbilder fest. Inzwischen wurde nun die MPEG-Spezifikation durch die "Moving Picture Expert Group" der ISO und des IEC aufgestellt, und sie legt Kompressions- und Expansionsalgorithmen für Farb-Bewegtbilder fest. Ferner wurde nun die H.261-Spezifikation vom CCITT aufgestellt, welche einen Kompressions- und Expansionsalgorithmus für Videokonferenz und Videofernsehen festlegt. Fig. 69 stellt ein Blockschaltbild dar, welches die Hauptprozesse in einem durch die internationale Standardspezifikation empfohlenen Bildkompressionsalgorithmus zeigt. Unter Bezugnahme auf Fig. 69 umfaßt der Bildkompressionsprozeß grundsätzlich ein voraussehendes Codieren, ein orthogonales Transformieren und ein Codieren mit variabler Länge. Wie aus der Fig. 69 zu ersehen ist, umfaßt jede der JPEG-, MPEG- und H.261-Spezifikationen beim Bildkompressionsprozeß ein diskretes Kosinustransformieren (nachstehend als "DCT" bezeichnet), eine Quantisierung sowie ein Huffman-Codieren.

Die JPEG-Spezifikation zur Verarbeitung eines Farb-Festbildes umfaßt als Grundsystem beispielsweise: adaptives DCT, Quantisierung, differentielle Pulscodemodulation (DPCM) und Huffman-Codierung. Die JPEG-Spezifikation umfaßt ein erweitertes System, adaptives DCT, hierarchische Kodierung, arithmetische Kodierung und adaptive Huffman-Kodierung.

Die MPEG-Spezifikation zur Bewegtbildspeicherung umfaßt eine Bewegungskompensation/Zwischenbildvoraussage, DCT, Quantisierung und Huffman-Codierung. Die H.261-Spezifikation für Videofernsehen und Videokonferenz umfaßt: Bewegungskompensation/Zwischenbildvoraussage, DCT, Quantisierung und Huffman-Codierung.

Eine LSI zur Bildkompression gemäß der vorstehenden internationalen Standardspezifikation ist entwickelt worden, und ein Entwurf derselben ist zum Beispiel in einem mit "LSI for Image Processing" benannten Artikel im Journal of the Institute of Television Engineers of Japan, Band 46, Nr. 3, Seiten 253—260, 1992 offenbart. Es wird angemerkt, daß die vorliegende Erfindung im allgemeinen bei einer Vorrichtung zur Bilddatenkompression, einer Vorrichtung zur Bilddatenexpansion und einem Bilddatenprozessor zur Bildkompression und/oder -expansion gemäß den vorstehend beschriebenen internationalen Standardspezifikationen verwendbar ist.

Fig. 70 stellt ein Blockschaltbild einer herkömmlichen Vorrichtung zur Bildkompression/-expansion dar. Unter Bezugnahme auf Fig. 70 umfaßt die Vorrichtung zur Bildkompression/-expansion einen Digitalsignalprozessor (nachstehend als "DSP" bezeichnet) 900 zur Bilddatenverarbeitung, einen Bilddatenspeicher 907, einen Codatenspeicher 908, einen Anzeigespeicher 909 und einen Arbeitsspeicher 910. Der DSP 900 ist mit den Speichern 907 bis 910 durch einen internen Bus 940 verbunden.

Fig. 71 stellt ein Blockschaltbild des in Fig. 70 gezeigten DSP 900 dar. Unter Bezugnahme auf Fig. 71 umfaßt der DSP 900 einen Anweisungsspeicher 901, eine Programmsteuereinrichtung 902, einen Datenspeicher 903, einen Datenoperator 904 und eine externe Schnittstelle 905. Ein Systembus 941 verbindet die Programmsteuereinrichtung 902' den Datenspeicher 903, den Datenoperator 904 und die externe Schnittstelle 905 miteinander. Im Betrieb wird ein im Anweisungsspeicher 901 gespeicherter (oder programmierter) Anweisungscode gelesen und der Programmsteuereinrichtung 902 zugeführt. Die Programmsteuereinrichtung 902 decodiert den zugeführten Anweisungscode und erzeugt verschiedene Steuersignale. In Reaktion auf die aus der Programmsteuereinrichtung 902 zugeführten Steuersignale empfängt der Datenoperator 904 durch die externe Schnittstelle 905 und den Datenspeicher 903 externe Daten und führt seine Operation aus. Das Ergebnis der Operation wird im Datenspeicher 903 gespeichert oder durch die externe Schnittstelle 905 ausgegeben. Auf diese Weise wird im DSP 900 der Prozeß gemäß dem im Anweisungsspeicher 901 gespeicherten Anweisungscode ausgeführt.

Ein Betrieb der Bilddatenkompression wird unter Bezugnahme auf die Fig. 70 und 71 beschrieben werden. Dabei wird angenommen, daß die zu komprimierenden Bilddaten im Bilddatenspeicher 907 gespeichert sind. Zuerst werden die Bilddaten, die im Bilddatenspeicher 907 gespeichert worden sind, in den Datenspeicher 903 im DSP 900 übertragen. Die im Datenspeicher 903 gespeicherten Bilddaten weisen ein Format auf, das durch den Datenoperator 904 umgewandelt wurde. Bei der Umwandlung werden abgetastete Daten des Farbdifferenzsignals oder RGB-Format-Daten zum Beispiel in YUV-Format-Daten umgewandelt. Die umgewandelten Daten werden im Datenspeicher 903 zeitweilig gespeichert. Der DSP 900 überträgt die umgewandelten Daten im Datenspeicher 903 in den Bilddatenspeicher 907.

Dann legt der DSP 900 unter Verwendung der im Bilddatenspeicher 907 gespeicherten Daten einen Bildkompressionsmodus fest. Zwischenbildkompressionsmodus, Innenbildkompressionsmodus und dergleichen sind als Bildkompressionsmodi bekannt. Im Zwischenbildkompressionsmodus wird manchmal ein Prozeß, wie beispielsweise eine Bewegungsvoraussage (oder -ermittlung), erforderlich. Berechnungsdaten zum Bestimmen des Kompressionsmodus und Berechnungsdaten zur Bewegungsermittlung werden zum Beispiel durch Ausführen einer Summation der Differenz-Absolutwerte der Bilddaten erhalten. Wenn eine derartige Operation ausgeführt wird, dann überträgt der DSP 900 die im Bilddatenspeicher 907 gespeicherten Daten in gleichartiger Weise wie bei der Formatumwandlung in den Datenspeicher 903, und der Datenoperator 904 verarbeitet die im Datenspeicher 903 gespeicherten Daten.

Auf Grundlage der durch die obengenannte Operation erhaltenen Berechnungsdaten legt der DSP 900 den Kompressionsmodus durch Bezugnahme auf im Arbeitsspeicher 910 gespeicherte Tabellendaten zur Modusbestimmung und/oder durch Ausführen einer Vergleichsoperation der Berechnungsdaten fest.

Nach der Festlegung des Kompressionsmodus führt der DSP 900 eine Umwandlungscodierung der im Bilddatenspeicher 907 gespeicherten Bilddaten durch. Das Umwandlungscodieren stellt beispielsweise das DCT dar. Der DSP 900 überträgt die im Bilddatenspeicher 907 gespeicherten Daten in den internen Speicher 903, und dann führt der Datenoperator 904 das DCT aus. Die verarbeiteten Daten werden im Datenspeicher 903 gehalten.

Ferner führt der DSP 900 ein Codieren mit variabler Länge (oder ein Entropiecodieren) aus. Das Codieren mit variabler Länge ist als Codeumwandlung bekannt, welche zum Beispiel den Huffman-Code verwendet. Der DSP 900 führt das Codieren mit variabler Länge der im Datenspeicher 903 gespeicherten Daten unter Bezugnahme auf eine im Arbeitsspeicher 910 gespeicherte Codeumwandlungstabelle aus. Nach der Codeumwandlung werden die verarbeiteten Daten aus dem Datenspeicher 903 in den Codedatenspeicher 903 übertragen und darin gespeichert.

Die Bilddatenverarbeitung zur Bildkompression wird in der vorstehend beschriebenen Weise ausgeführt, und die Bilddatenverarbeitung zur Bildexpansion wird ebenfalls in gleichartiger Weise durchgeführt. Insbesondere werden die zu expandierenden Daten dem Datenspeicher 903 im DSP 900 einmal zugeführt, und die Prozesse zur Bildexpansion werden durch den Datenoperator 904 an den gespeicherten Daten ausgeführt.

Wie vorstehend beschrieben, werden in der herkömmlichen Vorrichtung zur Bildkompression/-expansion die Bilddatenkompression und -expansion durch den DSP 900 mit einem einzigen Datenoperator 904 durchgeführt, und daher benötigt die Datenverarbeitung viel Zeit. Da der DSP 900 ferner nur ein Eingangs-/Ausgangsport (das heißt eine externe Schnittstelle 905) für die Daten aufweist, kann die Datenübertragung in der Vorrichtung zur Bildkompression/-expansion nicht wirksam ausgeführt werden, wenn eine große Menge von Bilddaten verarbeitet wird.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, eine Vorrichtung und ein Verfahren zur Bilddatenverarbeitung vorzusehen, welches zur Hochgeschwindigkeits-Verarbeitung geeignet ist.

Eine andere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung und ein Verfahren zur Bilddatenverarbeitung vorzusehen, durch welche die Bilddatenkompression mit hoher Geschwindigkeit ausgeführt werden kann.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung und ein Verfahren zur Bilddatenverarbeitung vorzusehen, durch welche die Bilddatenexpansion mit hoher Geschwindigkeit ausgeführt werden kann.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung zur Bilddatenverarbeitung vorzusehen, welche vorbestimmte Prozesse zur Bilddatenkompression und/oder Bilddatenexpansion und zur Bilddatenübertragung bei hoher Geschwindigkeit parallel ausführen kann.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung zur Bilddatenverarbeitung vorzusehen, welche dazu in der Lage ist, den Zugriff auf eine Mehrzahl von Speichereinheiten separat zu steuern.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung zur Bilddatenverarbeitung vorzusehen, welche die Bestimmung beim Auswählen entweder eines Codes variabler Länge oder eines Codes festgelegter Länge vereinfachen kann.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung zur Bilddatenverarbeitung vorzusehen, welche die Bestimmung beim Auswählen entweder von Laufdaten und Pegeldaten oder von decodierten Daten vereinfachen kann.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung zur Bilddatenverarbeitung vorzusehen, welche auf Pixeldaten eines gewünschten Gebiets auf dem Bildschirm in einem kurzen Zeitabschnitt zugreifen kann.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, ein Verfahren zur Bilddatenverarbeitung vorzusehen, bei welchem die Anzahl von Pixeldatenübertragungen verkleinert werden kann.

Gemäß einem Aspekt der vorliegenden Erfindung umfaßt die Vorrichtung zur Bilddatenverarbeitung einen Steuerprozessor zum Steuern einer Pipelineverarbeitung einer Mehrzahl von vorbestimmten Prozessen zur Bildkompression und/oder Bilddatenexpansion und einen Bildprozessor zum Ausführen eines vorbestimmten Prozesses der Mehrzahl von vorbestimmten Prozessen in Reaktion auf ein aus dem Steuerprozessor angelegtes Steuersignal.

Durch den vorstehend beschriebenen Aufbau steuert der Steuerprozessor die Pipelineverarbeitung einer Mehrzahl von vorbestimmten Prozessen zur Bilddatenkompression und/oder Bilddatenexpansion, so daß die Bilddaten mit hoher Geschwindigkeit verarbeitet werden können.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem anderen Aspekt der vorliegenden Erfindung umfaßt einen Steuerprozessor zum Steuern eines Parallelbetriebs von vorbestimmten Prozessen zur Bilddatenkompression und/oder zur Bilddatenexpansion und zur Bilddatenübertragung und eine Mehrzahl von Schaltungen zur Ausführungsanweisung, von denen jede individuell die Ausführung der vorbestimmten Prozesse sowie die Bilddatenübertragung in Reaktion auf ein aus dem Steuerprozessor angelegtes Steuersignal anweist.

Durch den vorstehend beschriebenen Aufbau steuert der Steuerprozessor den Parallelbetrieb einer Mehrzahl von vorbestimmten Prozessen zur Bilddatenkompression und/oder Bilddatenexpansion und zur Bilddatenübertragung. In Reaktion auf ein Steuersignal aus dem Steuerprozessor weist eine Mehrzahl von Schaltungen zur Ausführungsanweisung die Ausführung dieser Operationen einzeln an, so daß die Bilddaten mit hoher Geschwindigkeit verarbeitet werden können.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem anderen Aspekt der vorliegenden Erfindung ist

dazu in der Lage, auf eine erste Speichereinheit, welche Tabellendaten zur Umwandlung speichert, auf eine zweite Speichereinheit, welche die zu verarbeitenden Bilddaten speichert, auf eine dritte Speichereinheit, welche die Bilddaten für eine Anzeige speichert, sowie auf eine vierte Speichereinheit, welche die umgewandelten Bilddaten speichert, zuzugreifen, und sie umfaßt eine erste Zugriffsschaltung zum Zugreifen auf die erste Speichereinheit, eine zweite Zugriffsschaltung zum Zugreifen auf die zweite Speichereinheit, eine dritte Zugriffsschaltung zum Zugreifen auf die dritte Speichereinheit, eine vierte Zugriffsschaltung zum Zugreifen auf die vierte Speichereinheit und eine Schaltung zur Zugriffssteuerung zum individuellen Steuern des Beginns des Zugreifens auf die ersten bis vierten Zugriffsschaltungen.

Beim vorstehend beschriebenen Aufbau steuert die Schaltung zur Zugriffssteuerung individuell den Start des Zugreifens auf die erste bis vierte Zugriffsschaltung. Daher kann auf die ersten bis vierten Speichereinheiten einzeln zugegriffen werden, wobei die zur Bilddatenverarbeitung verwendeten Daten mit hoher Geschwindigkeit übertragen werden können, und daher können die Bilddaten mit hoher Geschwindigkeit verarbeitet werden.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem anderen Aspekt der vorliegenden Erfindung umfaßt eine Speichereinheit zum Speichern von einem Code von variabler Länge, welcher einer Kombination aus Laufdaten und Pegeldaten entspricht, von deren Codelängendaten und von einem Flag zur Verarbeitung eines Codes festgelegter Länge; eine Leseschaltung zum Lesen des Codes variabler Länge, der Codelängendaten desselben und des Flags zur Verarbeitung eines Codes festgelegter Länge, wie es den zugeführten Laufdaten und den zugeführten Pegeldaten entspricht; eine Schaltung zum Codieren einer festgelegten Länge zum Ausführen des Codierens an den zugeführten Laufdaten und den zugeführten Pegeldaten gemäß einer vorbestimmten Regel und zum Vorsehen eines entsprechenden Codes festgelegter Länge; eine Auswahlsschaltung, die auf das aus der Leseschaltung ausgegebene Flag zur Verarbeitung eines Codes festgelegter Länge reagiert, um entweder den Code variabler Länge oder den Code festgelegter Länge auszuwählen; und eine Codekoppelschaltung zum fortlaufenden Koppeln der aus der Auswahlsschaltung ausgegebenen Codes.

Im vorstehend beschriebenen Aufbau wählt die Auswahlsschaltung in Reaktion auf das aus der Leseschaltung zur Verfügung gestellte Flag zur Verarbeitung eines Codes festgelegter Länge entweder den Code variabler Länge oder den Code festgelegter Länge aus, so daß die Bestimmung gemäß dem Programm erheblich vereinfacht werden kann und die Bilddaten mit hoher Geschwindigkeit verarbeitet werden können.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung umfaßt eine Speichereinheit zum Speichern von einem Code variabler Länge, entsprechenden Laufdaten und Pegeldaten sowie eines Flags zur Verarbeitung einer festgelegten Länge; eine Gewinnungsschaltung zum Gewinnen eines Codes variabler Länge aus einem angelegten Code variabler Länge, welcher gekoppelte Codes variabler Länge umfaßt; eine Leseschaltung zum Lesen der entsprechenden Laufdaten, Pegeldaten und des Flags zur Verarbeitung einer festgelegten Länge aus der Speichereinheit entsprechend dem durch die Gewinnungsschaltung gewonnenen Codes variabler Länge; eine Schaltung zum Decodieren mit festgelegter Länge zum Decodieren des durch die Gewinnungsschaltung gemäß einer vorbestimmten Regel zum Vorsehen entsprechender decodierter Daten gewonnenen Codes variabler Länge; und eine Auswahlsschaltung, um in Reaktion auf das aus der Leseschaltung ausgegebene Flag einer festgelegten Länge entweder die Laufdaten und Pegeldaten oder die Codedaten auszuwählen.

Im vorstehend beschriebenen Aufbau wählt die Auswahlsschaltung in Reaktion auf das aus der Leseschaltung zur Verfügung gestellte Flag zur Verarbeitung einer festgelegten Länge entweder die Laufdaten und die Pegeldaten oder die decodierten Daten aus, so daß die Bestimmung gemäß dem Programm wesentlich vereinfacht werden kann und die Bilddaten mit hoher Geschwindigkeit verarbeitet werden können.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung umfaßt  $n$  ( $n \geq 2$ ) Speicherzellenanordnungen zum Speichern der Bilddaten, um die Pixel auf einem Bildschirm festzulegen, wobei die  $n$  Speicherzellenanordnungen  $n$  Pixeldaten vorsehen, von denen jede in Reaktion auf das Anlegen eines Zeilenadresssignals und eines Spaltenadresssignals in einer der entsprechenden  $n$  Speicherzellenanordnungen gespeichert wird, wobei die  $n$  Pixeldaten eine Mehrzahl von Pixeln festlegen, die in vertikaler Richtung auf dem Bildschirm angeordnet sind, und die Vorrichtung umfaßt ferner eine Schaltung zur Adresssignalerzeugung zum seriellen Ausgeben der Spaltenadresssignale, um die Pixeldaten der in der horizontalen Richtung auf dem Bildschirm angeordneten Pixel in Reaktion auf ein serielles Uhrsignal festzulegen.

Im vorstehend beschriebenen Aufbau legen  $n$  Pixeldaten, die in Reaktion auf ein Zeilenadresssignal und ein Spaltenadresssignal aus  $n$  Speicherzellenanordnungen gelesen werden, eine Mehrzahl von Pixeln fest, welche in der vertikalen Richtung auf dem Bildschirm angeordnet sind. Da die Schaltung zur Adresssignalerzeugung die Spaltenadresssignale in Reaktion auf ein serielles Uhrsignal seriell ausgibt, kann auf die Pixeldaten eines gewünschten Gebiets auf dem Bildschirm in einem kurzen Zeitabschnitt zugegriffen werden. Somit können die Bilddaten mit hoher Geschwindigkeit verarbeitet werden.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung umfaßt  $n$  ( $n \geq 2$ ) Speicherzellenanordnungen zum Speichern der Pixeldaten, um die Pixel auf einem Bildschirm festzulegen, wobei die  $n$  Speicherzellenanordnungen  $n$  Pixeldaten vorsehen, von denen jede in Reaktion auf das Anlegen eines Zeilenadresssignals und eines Spaltenadresssignals in einer der entsprechenden  $n$  Speicherzellenanordnungen gespeichert wird, wobei die  $n$  Pixeldaten eine Mehrzahl von Pixeln festlegen, die in horizontaler Richtung auf dem Bildschirm angeordnet sind, und die Vorrichtung umfaßt ferner eine Schaltung zur Adresssignalerzeugung zum seriellen Ausgeben der Spaltenadresssignale, um die Pixeldaten der in der vertikalen Richtung auf dem Bildschirm angeordneten Pixel in Reaktion auf ein serielles Uhrsignal festzulegen.

Im vorstehend beschriebenen Aufbau legen  $n$  Pixeldaten, die in Reaktion auf ein Zeilenadresssignal und ein Spaltenadresssignal aus  $n$  Speicherzellenanordnungen gelesen werden, eine Mehrzahl von Pixeln fest, welche in der horizontalen Richtung auf dem Bildschirm angeordnet sind. Da die Schaltung zur Adresssignalerzeugung die Spaltenadresssignale in Reaktion auf ein serielles Uhrsignal seriell ausgibt, kann auf die Pixeldaten eines ge-

wünschten Gebiets auf dem Bildschirm in einem kurzen Zeitabschnitt zugegriffen werden. Daher können die Bilddaten mit hoher Geschwindigkeit verarbeitet werden.

Die Vorrichtung zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung umfaßt: eine erste Pixeldaten-Speichereinheit zum Speichern von zu komprimierenden Pixeldaten; einen ersten Pixeldatenbus zum Übertragen der in der ersten Pixeldaten-Speichereinrichtung gespeicherten Pixeldaten; einen Pixelprozessor, welcher die durch den ersten Pixeldatenbus übertragenen Pixeldaten empfängt und unter Verwendung der übertragenen Pixeldaten Referenzdaten zur Bewegungsvoraussage gewinnt und welcher erste und zweite Datenports zum Vorsehen der Referenzdaten zur Bewegungsvoraussage aufweist; einen zweiten Pixeldatenbus zum Übertragen der Referenzdaten zur Bewegungsvoraussage, welche aus dem zweiten Datenport des Pixelprozessors ausgegeben werden; eine zweite Pixeldaten-Speichereinheit zum Speichern der durch den zweiten Datenbus übertragenen Referenzdaten zur Bewegungsvoraussage; eine Einheit zur Bewegungsvoraussage zum Ausführen der Bewegungsvoraussage durch Verwenden der Referenzdaten zur Voraussage, welche in der zweiten Pixeldaten-Speichereinheit gespeichert sind und durch den zweiten Pixeldatenbus zum Erzeugen eines Bewegungsvektors übertragen werden; und einen Prozessor zur Pipelinesteuerung zum Steuern der Pipelineverarbeitung des ersten und des zweiten Pixeldatenbusses, des Pixelprozessors und der Verarbeitungseinheit zur Bewegungsvoraussage.

Im vorstehend beschriebenen Aufbau stellt der Pixelprozessor die Referenzdaten zur Bewegungsvoraussage durch das erste und das zweite Datenport zur Verfügung, welche Daten in der entsprechenden ersten und zweiten Pixeldaten-Speichereinheit gespeichert werden, wobei die in der zweiten Pixeldaten-Speichereinheit gespeicherten Referenzdaten zur Bewegungsvoraussage durch den zweiten Pixeldatenbus in die Einheit zur Bewegungsvoraussage übertragen werden und die Einheit zur Bewegungsvoraussage die Bewegungsvoraussage ausführt und einen Bewegungsvektor erzeugt.

Im Ergebnis können die Daten bei wirksam verteilter Datenmenge effektiv übertragen werden, wobei die durch einen Pfad übertragene Datenmenge verkleinert werden kann, so daß die tatsächliche Datenübertragungsgeschwindigkeit verbessert werden kann und die Bilddaten mit hoher Geschwindigkeit verarbeitet werden können.

Das Verfahren zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung codiert eine Mehrzahl von Makroblockdaten, welche Pixel festlegen, die durch eine Pipelineverarbeitung eine Bildebene bilden, welches Verfahren umfaßt: Schritte zum Ausführen einer diskreten Kosinsumwandlung und einer Quantisierung der Makroblockdaten zum Erzeugen eines Pixelcodes, zum Ausführen eines Codierens mit variabler Länge an dem im vorstehend genannten Schritt erzeugten Pixelcode, um einen Code variabler Länge zu erzeugen, und zum Koppeln des im vorhergehenden Schritt erzeugten Codes variabler Länge, um eine Codefolge variabler Länge zu erzeugen. Durch das vorstehende Verfahren können alle Verarbeitungen der Mehrzahl von Makroblockdaten in einer pipelineartigen Weise ausgeführt werden, so daß die Bilddatenkompression mit hoher Geschwindigkeit ausgeführt werden kann.

Das Verfahren zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung decodiert eine Codefolge variabler Länge, welche eine Mehrzahl von Codes variabler Länge umfaßt, die Pixel festlegen, die bei der Pipelineverarbeitung eine Bildebene bilden, welches Verfahren Schritte zum Gewinnen des Codes variabler Länge aus der Codefolge variabler Länge, zum Ausführen eines Decodierens mit variabler Länge an dem im vorhergehenden Schritt gewonnenen Code variabler Länge, um einen Pixelcode zu erzeugen, und zum Ausführen einer inversen Quantisierung und einer inversen diskreten Kosinsumwandlung des im vorhergehenden Schritt erzeugten Pixelcodes umfaßt, um Makroblockdaten zu erzeugen, welche die die Bildebene bildenden Pixel festlegen.

Durch das vorstehende Verfahren kann eine eine Mehrzahl von Codes variabler Länge umfassende Codefolge variabler Länge durch Pipelineverarbeitung decodiert werden, und die Bilddaten können daher mit hoher Geschwindigkeit expandiert werden.

Das Verfahren zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung dient dem Abbilden der ersten Pixeldaten in eine Speichereinheit, die eine Mehrzahl von ersten Pixeldaten speichert, welche die Pixel eines Bildschirms festlegen, welches Verfahren umfaßt einen ersten Schritt zum Bestimmen einer Mehrzahl von zweiten Pixeldaten, die zweidimensional zueinander benachbart sind und in einem rechteckigen Gebiet aus den ersten Pixeldaten enthalten sind, einen zweiten Schritt zum Zuweisen einer Adresse der Speichereinheit an die zweiten Pixeldaten, wobei die zweiten Pixeldaten als ein Datenstück betrachtet werden, und einen dritten Schritt zum Übertragen der zweiten Daten aus oder zu der Speichereinheit unter Verwendung der Adresse der zweiten Pixeldaten. Daher wird im vorstehend genannten Verfahren zur Bilddatenverarbeitung eine Mehrzahl von zweiten Pixeldaten, die zweidimensional zueinander benachbart und in einem rechteckigen Gebiet enthalten sind, als ein Datenstück betrachtet und durch Zuweisen einer Adresse übertragen, so daß die Anzahl von Übertragungen zum Übertragen der ersten Pixeldaten im rechteckigen Gebiet verkleinert werden kann. Daher können die Bilddaten mit hoher Geschwindigkeit verarbeitet werden.

Das Verfahren zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung dient dem Abbilden von ersten und zweiten Farbdifferenz-Pixeldaten auf eine Speichereinheit, welche die ersten und zweiten Farbdifferenz-Pixeldaten speichert, die miteinander in Beziehung stehen, um auf einem Bildschirm Pixel festzulegen, welches Verfahren umfaßt: einen ersten Schritt zum abwechselnden Abbilden von Übertragungseinheiten der ersten und der zweiten Farbdifferenz-Pixeldaten auf die Speichereinheit und einen zweiten Schritt zum Übertragen der ersten oder der zweiten Farbdifferenz-Pixeldaten zu oder aus der Speichereinheit durch die Übertragungseinheit.

Daher werden beim vorstehend beschriebenen Verfahren zur Bilddatenverarbeitung in Beziehung stehende erste und zweite Farbdifferenz-Pixeldaten durch die Übertragungseinheit abwechselnd abgebildet und die ersten oder zweiten Farbdifferenz-Pixeldaten durch diese Einheit übertragen. Daher können die ersten und die

zweiten Farbdifferenz-Pixeln eines vorgeschriebenen Gebiets durch Erzeugen einer Adresse übertragen werden. Im Ergebnis können die Bilddaten mit hoher Geschwindigkeit verarbeitet werden.

Das Verfahren zur Bilddatenverarbeitung gemäß einem weiteren Aspekt der vorliegenden Erfindung dient dem Abbilden von ersten und zweiten Farbdifferenz-Pixeln auf eine Speichereinheit, die die ersten und zweiten Farbdifferenz-Pixeln speichert, welche miteinander in Beziehung stehen, um auf einem Bildschirm Pixel festzulegen, welches Verfahren umfaßt: einen ersten Schritt zum Abbilden von dritten Farbdifferenz-Pixeln, welche die gleiche Anzahl der ersten und zweiten Farbdifferenz-Pixeln umfaßt, als ein Datenstück auf die Speichereinrichtung, und einen zweiten Schritt zum Übertragen der dritten Farbdifferenz-Pixeln als ein Datenstück aus oder zu der Speichereinheit.

Daher werden beim vorstehend beschriebenen Verfahren zur Bilddatenverarbeitung die dritten Farbdifferenz-Pixeln, welche die gleiche Anzahl der miteinander in Beziehung stehenden ersten und zweiten Farbdifferenz-Pixeln umfassen, als ein Datenstück auf die Speichereinrichtung abgebildet, und die dritten Farbdifferenz-Pixeln werden als ein Datenstück übertragen. Daher können die ersten und die zweiten Farbdifferenz-Pixeln eines vorgeschriebenen Gebiets durch Erzeugen einer Adresse übertragen werden. Folglich können die Bilddaten mit hoher Geschwindigkeit verarbeitet werden.

Die vorstehenden und weitere Aufgaben, Merkmale, Aspekte und Vorteile der vorliegenden Erfindung werden aus der folgenden detaillierten Beschreibung der vorliegenden Erfindung augenscheinlicher werden, wenn sie in Verbindung mit den beigefügten Figuren zur Kenntnis genommen wird.

Von den Figuren zeigen:

Fig. 1 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welche eine erste Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 2 ein Blockschaltbild, welches die Prozesse zur Bildkompression bei der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion zeigt;

Fig. 3 ein Blockschaltbild, welches die Prozesse zur Bildexpansion durch die in Fig. 1 dargestellte Vorrichtung zur Bildkompression/-expansion zeigt;

Fig. 4 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welche eine zweite Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 5 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welche eine dritte Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 6 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welche eine vierte Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 7 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welche eine fünfte Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 8 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welche eine sechste Ausführungsform der vorliegenden Erfindung zeigt;

Fig. 9 ein Blockschaltbild, welches ein Beispiel einer Steuereinheit zeigt;

Fig. 10 ein Blockschaltbild, welches ein anderes Beispiel der Steuereinheit zeigt;

Fig. 11 ein Zeitablaufdiagramm, welches die Pipelineverarbeitung zeigt, die unter der Steuerung eines in Fig. 9 dargestellten Gesamtsteuerungsprozessors ausgeführt wird;

Fig. 12 Paralleloperationen in einem in Fig. 11 gezeigten Zeitabschnitt T10;

Fig. 13 ein Blockschaltbild eines in Fig. 9 dargestellten Prozessors zur Gesamtsteuerung;

Fig. 14 ein Blockschaltbild einer in Fig. 9 dargestellten Steuereinheit zur Bewegungsvoraussage;

Fig. 15 ein Blockschaltbild einer in Fig. 9 dargestellten Einheit zur Bildformat-Umwandlung;

Fig. 16 ein Blockschaltbild der in Fig. 9 dargestellten Steuereinheit zur Bilddatenübertragung;

Fig. 17 ein Blockschaltbild einer in Fig. 9 dargestellten Einheit zur Anweisungsübertragung;

Fig. 18 ein Blockschaltbild, welches ein Beispiel einer Pixelverarbeitungseinheit zeigt;

Fig. 19 ein detailliertes Blockschaltbild der in Fig. 18 dargestellten Pixelverarbeitungseinheit;

Fig. 20 ein Blockschaltbild, welches ein anderes Beispiel der Pixelverarbeitungseinheit zeigt;

Fig. 21 ein detailliertes Blockschaltbild der in Fig. 20 dargestellten Pixelverarbeitungseinheit;

Fig. 22 ein Flußdiagramm, das den Codierbetrieb (die Innenbildvoraussage) bei der in Fig. 19 dargestellten Pixelverarbeitungseinheit zeigt;

Fig. 23 ein Flußdiagramm, welches den Decodierbetrieb (die Innenbildvoraussage) in der in Fig. 19 dargestellten Pixelverarbeitungseinheit zeigt;

Fig. 24 ein Flußdiagramm des Codierbetriebs (Zwei-Weg-Voraussage) bei der in Fig. 19 dargestellten Pixelverarbeitungseinheit;

Fig. 25 ein Flußdiagramm des Decodierbetriebs (Zwei-Wege-Voraussage) bei der in Fig. 19 dargestellten Pixelverarbeitungseinheit;

Fig. 26 ein Zeitablaufdiagramm, das die Pipelineverarbeitung zum Codieren bei der in Fig. 19 dargestellten Pixelverarbeitungseinheit zeigt;

Fig. 27 ein Zeitablaufdiagramm, das die Pipelineverarbeitung bei der Innenbildvoraussage bei der in Fig. 19 dargestellten Pixelverarbeitungseinheit zeigt;

Fig. 28 ein Zeitablaufdiagramm, das die Pipelineverarbeitung bei der Ein-Weg-Verarbeitung in der in Fig. 19 dargestellten Pixelverarbeitungseinheit zeigt;

Fig. 29 ein Zeitablaufdiagramm, das die Pipelineverarbeitung bei der Zwei-Weg-Verarbeitung in der in Fig. 19 dargestellten Pixelverarbeitungseinheit zeigt;

Fig. 30 ein Blockschaltbild, welches einen ersten Systemaufbau zur Verarbeitung einer variablen Länge zeigt;

Fig. 31 ein Flußdiagramm des Codierens mit variabler Länge bei dem in Fig. 30 dargestellten ersten Systemaufbau;



Fig. 32 ein Flußdiagramm des Decodierens mit variabler Länge bei dem in Fig. 30 dargestellten ersten Systemaufbau;

Fig. 33 einen Datenspeicheraufbau eines Tabellenspeichers für das in Fig. 30 dargestellte Codieren einer variablen Länge;

Fig. 34 einen Datenspeicheraufbau eines Tabellenspeichers für das in Fig. 30 dargestellte Decodieren mit variabler Länge;

Fig. 35 ein Flußdiagramm, welches einen Prozeß zum Erzeugen einer Code folge variabler Länge darstellt;

Fig. 36 ein Blockschaltbild, das ein erstes Beispiel eines Prozessors für variable Längen zeigt;

Fig. 37 ein Blockschaltbild, das ein zweites Beispiel eines Prozessors für variable Längen zeigt;

Fig. 38 ein Blockschaltbild, das ein drittes Beispiel eines Prozessors für variable Längen zeigt;

Fig. 39 ein Blockschaltbild, das einen zweiten Systemaufbau für die Verarbeitung einer variablen Länge zeigt;

Fig. 40 ein Blockschaltbild des in Fig. 39 dargestellten Prozessors für variable Längen;

Fig. 41 ein Blockschaltbild des in Fig. 40 dargestellten Adreßgenerators;

Fig. 42 ein Flußdiagramm des Codierens einer variablen Länge in dem in Fig. 41 dargestellten Adreßgenerator;

Fig. 43 ein Flußdiagramm des Decodierens mit variabler Länge im in Fig. 41 dargestellten Adreßgenerator;

Fig. 44 ein Blockschaltbild der in Fig. 40 dargestellten Schaltung zum Erzeugen/Trennen einer Codefolge variabler Länge;

Fig. 45 ein Flußdiagramm des Codierens mit variabler Länge in der in Fig. 44 dargestellten Schaltung zum Erzeugen/Trennen einer Code folge variabler Länge;

Fig. 46 ein Flußdiagramm des Decodierens mit variabler Länge in der in Fig. 44 dargestellten Schaltung zum Erzeugen/Trennen einer Codefolge variabler Länge;

Fig. 47 ein Blockschaltbild einer in Fig. 40 dargestellten externen Schnittstellenschaltung;

Fig. 48 ein Flußdiagramm des Codierens mit variabler Länge bei der in Fig. 47 dargestellten externen Schnittstellenschaltung;

Fig. 49 ein Flußdiagramm des Decodierens mit variabler Länge bei der in Fig. 47 dargestellten externen Schnittstellenschaltung;

Fig. 50 eine Tabelle aus Gruppen von Codes variabler Länge, welche im in Fig. 39 dargestellten Prozessor variable Längen verwendet wird;

Fig. 51 das Datenformat der Daten, die im Tabellenspeicher für das in Fig. 39 dargestellte Codieren mit variabler Länge gespeichert sind;

Fig. 52 das Datenformat der beim Codieren mit variabler Länge zu codierenden Daten;

Fig. 53 den Grundaufbau der Speicherzellenanordnung in einem Bildpufferspeicher;

Fig. 54 das Blockschaltbild eines Bildpufferspeichers;

Fig. 55 einen Systemaufbau des Bildpufferspeichers, der DRAMs verwendet;

Fig. 56 eine Pixelanordnung, welche die Beziehung zwischen den Daten, die im in Fig. 55 dargestellten Bildpufferspeicher gespeichert sind, und den Pixeln auf dem Bildschirm zeigt;

Fig. 57 eine Pixelanordnung, welche die Beziehung zwischen den Daten, die im in Fig. 60 dargestellten Bildpufferspeicher gespeichert sind, und den Pixeln auf dem Bildschirm zeigt;

Fig. 58 eine Pixelanordnung, welche das Adressieren durch eine in Fig. 55 dargestellte DRAM-Steuereinrichtung darstellt;

Fig. 59 ein Zeitablaufdiagramm, welches den Lesebetrieb eines Bildpufferspeichers unter der Steuerung der in Fig. 55 dargestellten DRAM-Steuereinrichtung zeigt;

Fig. 60 einen Systemaufbau eines Bildpufferspeichers, der SDRAMs verwendet;

Fig. 61 ein Zeitablaufdiagramm, das den Lesebetrieb des SDRAMs zeigt;

Fig. 62 ein Beispiel des Betriebs des in Fig. 60 dargestellten Adreßgenerators;

Fig. 63 ein Zeitablaufdiagramm, welches ein Beispiel des Betriebs des in Fig. 60 dargestellten Adreßgenerators zeigt;

Fig. 64 ein Zeitablaufdiagramm, welches ein anderes Beispiel des Betriebs des in Fig. 60 dargestellten Adreßgenerators zeigt;

Fig. 65 ein Zeitablaufdiagramm, welches ein weiteres Beispiel des Betriebs des in Fig. 60 dargestellten Adreßgenerators zeigt;

Fig. 66 ein Blockschaltbild des in Fig. 60 dargestellten Adreßgenerators;

Fig. 67 eine Adresse-Seite-Entsprechung, welche eine Beziehung zwischen Seiten sowie Zeilen- und Spaltenadressen zeigt, welche von dem in Fig. 66 dargestellten Adreßgenerator vorgesehen wird;

Fig. 68 eine Seite-Pixel-Entsprechung, welche eine in Fig. 67 dargestellte Beziehung zwischen den Seiten und den Pixeln auf dem Bildschirm zeigt;

Fig. 69 ein Blockschaltbild, welches Hauptprozesse gemäß einem durch eine internationale Standardspezifikation empfohlenen Bildkompressionsalgorithmus zeigt;

Fig. 70 ein Blockschaltbild einer herkömmlichen Vorrichtung zur Bildkompression/-expansion;

Fig. 71 ein Blockschaltbild des in Fig. 70 dargestellten DSP;

Fig. 72 eine Darstellung eines ersten Verfahrens zum Abbilden von Pixeldaten auf den Bildschirm an einem Bildpufferspeicher;

Fig. 73 eine erste Darstellung der Übertragung eines Datengebiets, welches 8 Pixel in der horizontalen Richtung und 8 Pixel in der vertikalen Richtung umfaßt;

Fig. 74 eine zweite Darstellung der Übertragung eines Datengebiets, welches 8 Pixel in der horizontalen Richtung und 8 Pixel in der vertikalen Richtung umfaßt;

Fig. 75 eine dritte Darstellung der Übertragung eines Datengebiets, welches 8 Pixel in der horizontalen

Richtung und 8 Pixel in der vertikalen Richtung umfaßt;

Fig. 76 eine vierte Darstellung der Übertragung eines Datengebiets, welches 8 Pixel in der horizontalen Richtung und 8 Pixel in der vertikalen Richtung umfaßt;

Fig. 77 eine Darstellung eines zweiten Verfahrens zum Abbilden von Pixeldaten auf dem Bildschirm am Bildpufferspeicher;

Fig. 78 ein erstes Verfahren zum Abbilden erster und zweiter Farbdifferenz-Pixeldaten auf einen Bildpufferspeicher;

Fig. 79 ein zweites Verfahren zum Abbilden erster und zweiter Farbdifferenz-Pixeldaten auf einen Bildpufferspeicher;

Fig. 80 ein drittes Verfahren zum Abbilden erster und zweiter Farbdifferenz-Pixeldaten auf einen Bildpufferspeicher;

Fig. 81 ein viertes Verfahren zum Abbilden erster und zweiter Farbdifferenz-Pixeldaten auf einen Bildpufferspeicher;

Fig. 82 einen Betrieb des Prozessors zur Gesamtsteuerung zur Zeit des Codierens der Zwei-Weg-Voraussage;

Fig. 83 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine siebente Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 84 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine achte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 85 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine neunte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 86 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine zehnte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 87 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine elfte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 88 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine zwölfte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 89 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine dreizehnte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 90 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine vierzehnte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 91 ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion, welches eine fünfzehnte Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 92 ein Blockschaltbild, welches ein weiteres Beispiel der Steuereinheit darstellt;

Fig. 93 ein Blockschaltbild, welches ein weiteres Beispiel der Pixelverarbeitungseinheit zeigt;

Fig. 94 ein erstes detailliertes Blockschaltbild der in Fig. 93 dargestellten Pixelverarbeitungseinheit; und

Fig. 95 ein zweites detailliertes Blockschaltbild der in Fig. 18 dargestellten Pixelverarbeitungseinheit.

Im folgenden werden Vorrichtungen zur Bildkompression/-expansion (oder Systeme zur Bildkompression/-expansion) beschrieben werden, welche erste bis fünfzehnte Ausführungsformen der vorliegenden Erfindung darstellen. Der Systemaufbau dieser Ausführungsformen wird zuerst beschrieben werden, gefolgt von einer Beschreibung des internen Systemaufbaus.

## 1. Systemaufbau

### Die erste Ausführungsform

Fig. 1 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 101 dar, welche eine erste Ausführungsform der vorliegenden Erfindung darstellt. Unter Bezugnahme auf Fig. 1 umfaßt die Vorrichtung zur Bildkompression/-expansion 101: eine Schaltung einer Host-Schnittstelle (I/F) 1 zum Eingeben/Ausgeben in und aus einem Host-Computer 93; eine Steuereinheit 2, welche zwei Prozessoren (nicht dargestellt) umfaßt; eine Pixelverarbeitungseinheit 3 zum DCT und zur Quantisierung; eine Einheit zur Bewegungsvoraussage (oder -ermittlung) 41; einen Bildpufferspeicher 51 zum Speichern von zu verarbeitenden Bilddaten; einen Pufferspeicher 6 zum Speichern von Codedaten; einen Arbeitsspeicher 7 zum Speichern von Tabellendaten, die für verschiedene Umwandlungen erforderlich sind; und einen Eingabe-/Ausgabespeicher 81 zum Speichern der Bilddaten aus einer TV-Kamera 91 und/oder zum Anlegen der gespeicherten Bilddaten an eine Anzeigevorrichtung (CRT) 92.

Ein Host-Bus HB weist eine Busbreite von 24 Bit (in der Figur durch "24b" bezeichnet) auf und ist zur Datenübertragung jeweils zwischen der Host-Schnittstellen-Schaltung 1, der Steuereinheit 2, der Pixelverarbeitungseinheit 3, dem Pufferspeicher 6 und dem Arbeitsspeicher 7 vorgesehen. Ein Pixeldatenbus PB weist eine Busbreite von 32 Bit (32b) auf und ist zur Datenübertragung jeweils zwischen der Steuereinheit 2, der Pixelverarbeitungseinheit 3, der Einheit zur Bewegungsvoraussage 41 und dem Bildpufferspeicher 51 vorgesehen. Ein Codedatenbus CB weist eine Busbreite von 16 Bit (16b) auf und ist zur Codedatenübertragung zwischen der Steuereinheit 2, der Pixelverarbeitungseinheit 3 und dem Pufferspeicher 6 vorgesehen. Ein Eingabe-/Ausgabebus IOB weist eine Busbreite von 24 Bit (24b) auf und ist zur Datenübertragung zwischen der Steuereinheit 2 und dem Eingabe-/Ausgabespeicher 81 vorgesehen.

Die Host-Schnittstellen-Schaltung 1 umfaßt eine Logikschaltung, welche durch eine programmierbare Logikeinrichtung gebildet wird, wie beispielsweise ein Allzweck-Logik-LSI (diskret), ein PLD oder FPGA. Der Bildpufferspeicher 51 ist vor allem zum zeitweiligen Speichern der zu komprimierenden Bilddaten und der zuzuweisenden Bilddaten vorgesehen. Als Bildpufferspeicher 51 wird ein Speicher mit großer Speicherkapazität

verwendet, wie beispielsweise ein statischer Speicher mit wahlfreiem Zugriff (SRAM), ein dynamischer Speicher mit wahlfreiem Zugriff (DRAM), ein synchroner DRAM und ein Cache-DRAM.

Der Pufferspeicher 6 ist zum zeitweiligen Speichern von Lauf-/Pegeldaten und Bitstromdaten vorgesehen, die beim Codieren der Bilder erhalten werden. Der Pufferspeicher 6 wird durch einen FIFO-Speicher (FIFO) gebildet. Es kann jedoch ein DRAM oder ein SRAM verwendet werden, wie es erforderlich ist.

Der Arbeitsspeicher 7 ist zum Speichern von Tabellendaten des DCT/inversen DCT, von Tabellendaten der Quantisierung/inversen Quantisierung, von Huffman-Tabellendaten zur Verarbeitung einer variablen Länge, eines Programms (Mikrocode) zum Verarbeiten in der Steuereinheit 2 und der Pixelverarbeitungseinheit 3, von Daten zur Initialisierung und so weiter vorgesehen. Der Arbeitsspeicher 7 wird durch SRAMs gebildet.

Der Eingabe-/Ausgabespeicher 81 ist zum Speichern der Bilddaten für die TV-Kamera 91 und/oder eine Anzeigevorrichtung 92 vorgesehen. Der Eingabe-/Ausgabespeicher 81 wird durch Video-RAMs gebildet.

Die Steuereinheit 2 umfaßt einen Mikroprozessor (nicht dargestellt) zur Gesamtsteuerung und einen Prozessor (nicht dargestellt) zum Verarbeiten variabler Längen. Der Prozessor zur Gesamtsteuerung steuert sowohl die Pipelineverarbeitung des DCT, der Quantisierung und des Codierens mit variabler Länge zur Bildkompression als auch die Pipelineverarbeitung des Decodierens mit variabler Länge, der inversen Quantisierung und des inversen DCT zur Bildexpansion.

Die Pixelverarbeitungseinheit 3 führt eine Pixelverarbeitung aus, wie beispielsweise das DCT und die Quantisierung bei der Bildkompression, und führt eine Pixelverarbeitung aus, wie beispielsweise die inverse Quantisierung und das inverse DCT bei der Bildexpansion.

Die Einheit zur Bewegungsvoraussage 41 führt einen Prozeß zur Bewegungsermittlung aus, wie beispielsweise Ein-Weg- und Zwei-Weg-Bildvoraussagen.

Fig. 2 stellt ein Schaltbild dar, welches Prozesse bei der Bildkompression in der in Fig. 1 gezeigten Vorrichtung zur Bildkompression/-expansion 101 zeigt. In Fig. 2 ist ein in der MPEG-Spezifikation aufgeführtes Codierverfahren beispielhaft dargestellt.

Unter Bezugnahme auf Fig. 2 werden die zu komprimierenden Bilddaten ID durch einen Subtrahierer 921 und einen Schalter 922 an einen DCT-Prozessor 923 gelegt. Nach dem DCT wird die Quantisierung durch einen Quantisierer 924 ausgeführt. Die quantisierten Daten werden an einen Codierer für variable Längen 925 gelegt und einem Codieren mit variabler Länge unterworfen, und die codierten Daten werden als komprimierte Bilddaten VD durch einen Puffer 926 bereitgestellt. Inzwischen wurden die quantisierten Daten auch an einen inversen Quantisierer sowie einen inversen DCT-Prozessor gelegt, und die inverse Quantisierung und das inverse DCT zur Bewegungsvoraussage werden ausgeführt. Die Ausgangsdaten aus dem inversen DCT-Prozessor werden an einen Addierer 929 gelegt.

Die Bilddaten ID werden ferner an eine Bewegungsvoraussage-Einrichtung 930 gelegt, und die vorausgesagten Werte werden an den Addierer 929 gelegt. Die Ausgangsdaten aus dem Addierer 929 werden durch einen Schalter 931 an den Subtrahierer 921 und die Bewegungsvoraussage-Einrichtung 930 gelegt. Durch einen Schaltbetrieb der Schalter 922 und 931 wird die Bildkompression ausgeführt, wobei die Bewegungsvoraussage Berücksichtigung findet.

Fig. 3 stellt ein Schaltbild dar, welches einen Bildexpansionsprozeß bei der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion 101 zeigt. Unter Bezugnahme auf Fig. 3 werden durch einen Puffer 911 Codedaten variabler Länge VD an einen Decodierer für variable Längen 912 gelegt. Nach der Decodierung mit variabler Länge werden die Codedaten an einen inversen Quantisierer 913 gelegt und einer inversen Quantisierung unterworfen. Nach der inversen Quantisierung wird durch einen inversen DCT-Prozessor 914 eine inverse DCT ausgeführt, und die verarbeiteten Daten werden an einen Addierer 915 gelegt. Der Addierer führt eine Addition der Ausgangsdaten durch, einschließlich derjenigen Daten, die zur Bewegungsvoraussage verwendet worden sind, und stellt expandierte Bilddaten ID zur Verfügung.

Die in den Fig. 2 und 3 dargestellten Prozesse sind grundlegend in der MPEG-Spezifikation aufgeführt. Diese Prozesse zur Bildkompression und -expansion werden unter der Steuerung einer Pipelineverarbeitung in der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion 101 wirksam ausgeführt, welche Pipelineverarbeitung nachstehend beschrieben werden wird.

#### Die zweite Ausführungsform

Fig. 4 zeigt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 102, welche eine zweite Ausführungsform der vorliegenden Erfindung darstellt. Im Vergleich zu der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion 101 ist die Einheit zur Bewegungsvoraussage 41 in der in Fig. 4 dargestellten Vorrichtung zur Bildkompression/-expansion 102 entfernt. Insbesondere kann in der in Fig. 4 gezeigten Ausführungsform sowohl ein Decodieren von Bewegtbildern als auch ein Codieren und Decodieren von Festbildern ausgeführt werden. Das Decodieren der Bewegtbilder ist ein inverser Prozeß des Codierens. Es umfaßt jedoch keine Bewegungsvoraussage, und alternativ wird eine Bewegungsdecodierung hinzugefügt, bei welcher das vorausgesagte Bild durch Empfangen eines Bewegungsvektors erzeugt wird.

#### Die dritte Ausführungsform

Fig. 5 stellt ein Blockschaltbild dar, das eine Vorrichtung zur Bildkompression/-expansion 103 zeigt, welche eine dritte Ausführungsform der vorliegenden Erfindung darstellt. Unter Bezugnahme auf Fig. 5 umfaßt die Vorrichtung zur Bildkompression/-expansion 103 im Vergleich zu der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion 101 einen mit einem Pixeldatenbus PB verbundenen Eingabe-/Ausgabespeicher 82. Insbesondere ist der Eingabe-/Ausgabespeicher 82 durch den Pixeldatenbus PB mit der Steuereinheit 2 verbun-

den. Folglich sind in der Steuereinheit 2 der Eingabe-/Ausgabespeicher 82 und der Bildpufferspeicher 6 auf einem gemeinsamen Adressenraum abgebildet, und daher kann die Anzahl von Eingangs-/Ausgangs-Schnittstellen-Schaltungen (Eingangs-/Ausgangsports) der Steuereinheit verkleinert werden.

5

#### Die vierte Ausführungsform

Fig. 6 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 104 dar, welche eine vierte Ausführungsform der vorliegenden Erfindung darstellt. Unter Bezugnahme auf Fig. 6 umfaßt die Vorrichtung zur Bildkompression/-expansion 104 im Vergleich zur ersten Vorrichtung zur Bildkompression/-expansion 101 eine Einheit zur Zwei-Weg-Bewegungsvoraussage 42, welche eine Bewegungsvoraussage in Vorwärts- und Rückwärtsrichtung parallel (oder gleichzeitig) ausführen kann. Insbesondere die Einheit zur Zwei-Weg-Bewegungsvoraussage 42 umfaßt eine Einheit zur Bewegungsvoraussage 43 zur Bewegungsvoraussage in der Vorwärtsrichtung und eine Einheit zur Bewegungsvoraussage 44 zur Bewegungsvoraussage in der Rückwärtsrichtung. Die Einheiten zur Bewegungsvoraussage 43 und 44 sind mit einem Pixeldatenbus PB verbunden. Da die Bewegungsvoraussage in Vorwärts- und Rückwärtsrichtung durch Verwenden der beiden Einheiten zur Bewegungsvoraussage 43 und 44 parallel oder gleichzeitig ausgeführt werden kann, kann die Zwei-Weg-Bewegungsvoraussage in einem kurzen Zeitabschnitt wirkungsvoll ausgeführt werden.

20

#### Die fünfte Ausführungsform

Fig. 7 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 105 dar, welche eine fünfte Ausführungsform der vorliegenden Erfindung darstellt.

Im Vergleich zu der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion 101 ist die Vorrichtung zur Bildkompression/-expansion 105 mit einer Einheit zur Bewegungsvoraussage 45 versehen, welche dazu in der Lage ist, eine Zwei-Weg-Bewegungsvoraussage gleichzeitig auszuführen, und mit zwei Bildpufferspeichern 52 und 53, von denen jeder mit einem Pixeldatenbus PB verbunden ist, versehen. Die Einheit zur Bewegungsvoraussage 45 kann die Voraussage in beiden Richtungen gleichzeitig ausführen, und daher kann die Zwei-Weg-Voraussage in einem Zeitabschnitt abgeschlossen werden, welcher entweder für die Vorwärts- oder die Rückwärtsvoraussage erforderlich ist. Die Einheit zur Bewegungsvoraussage 45 weist zwei Eingangs-/Ausgangsports auf, von denen der eine mit dem Pixeldatenbus PB verbunden ist und der andere durch einen lokalen Bus LB1 mit dem Bildpufferspeicher 53 verbunden ist. Einer der beiden Bildpufferspeicher 52 und 53 wird als Bildspeicher zur Voraussage einer Vorwärtsbewegung verwendet, während der andere Bildpufferspeicher als Bildspeicher zur Voraussage einer Rückwärtsbewegung verwendet wird. Referenz-Bilddaten werden zum Pixeldatenbus PD und zum lokalen Bus LB1 übertragen. Musterbilddaten und Daten des Bewegungsvektors werden aus der Einheit zur Bewegungsvoraussage 45 durch einen zweiten lokalen Bus LB2 zu einer Steuereinheit 2 übertragen.

35

#### Die sechste Ausführungsform

Fig. 8 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 106 dar, welche eine sechste Ausführungsform der vorliegenden Erfindung darstellt. Im Vergleich zu der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion 101 umfaßt die Vorrichtung zur Bildkompression/-expansion 106 eine Einheit zur Bewegungsvoraussage 46 mit zwei Eingangs-/Ausgangsports. Das erste Eingangs-/Ausgangs-  
port der Einheit zur Bewegungsvoraussage 46 ist mit einem Pixeldatenbus PB verbunden, und ihr zweites  
Eingangs-/Ausgangs-  
port ist durch einen lokalen Bus LB3 mit einer Steuereinheit 2 verbunden. Die Einheit zur Bewegungsvoraussage 46 legt die durch eine Bewegungsvoraussage erhaltenen Bewegungsvektor-Daten durch den lokalen Bus LB3 direkt an die Steuereinheit 2. Da die Bewegungsvektor-Daten aus Signalen zu 8 Bit bestehen, weist der lokale Bus LB3 eine Busbreite von 8 Bit auf. Da die Bewegungsvektor-Daten durch den lokalen Bus LB3 direkt an die Steuereinheit 2 gelegt werden, kann die Parallelverarbeitung in der Vorrichtung zur Bildkompression/-expansion 106 verbessert werden, und im Ergebnis kann die Bildkompression/-expansion effizienter mit hoher Geschwindigkeit ausgeführt werden.

Die in den Fig. 4 bis 8 dargestellten Vorrichtungen zur Bildkompression/-expansion 102 bis 106 weisen grundsätzlich die gleiche Betriebsweise wie die in Fig. 1 gezeigte Vorrichtung zur Bildkompression/-expansion 101 auf. Insbesondere wird in einer beliebigen der zweiten bis sechsten Ausführungsformen die in den Fig. 2 und 3 dargestellte Bildkompression/-expansion unter der Steuerung für eine Pipelineverarbeitung ausgeführt. Daher wird bemerkt, daß die Bildkompression/-expansion mit hoher Geschwindigkeit effizient ausgeführt werden kann. Einzelheiten der Pipelineverarbeitung werden nachstehend unter Bezugnahme auf die Fig. 11 und 12 beschrieben werden.

Die Bilddatenübertragung bei der Bildcodierung zur Datenübertragung durch den Pixeldatenbus PB in der Vorrichtung zur Bildkompression/-expansion 106 der sechsten Ausführungsform und die Menge der Bilddatenübertragung zur Verarbeitung der Bilddaten von einem Makroblock werden im folgenden genau beschrieben werden.

Es wird angenommen, daß die zu codierenden Bilddaten bereits in einem Bildpufferspeicher 51 gespeichert worden sind. Die Datenübertragungsmenge wird unter der Voraussetzung berechnet, daß das Bildformat 4 : 2 : 0 beträgt, der Bereich der Bewegungsermittlung  $\pm 16$  Pixel umfaßt und daß die Struktur eine Bildstruktur aufweist.

Jede Einheit bewirkt eine Eingabe und eine Ausgabe von Bilddaten in und aus dem Pixeldatenbus PB unter der Steuerung einer Steuereinheit 2. Das Betriebs-Timing jeder Einheit wird durch die Steuereinheit 2 gesteuert.

Zuerst werden durch den Pixeldatenbus PB Originalbilddaten aus der Steuereinheit 2 in den Bildpufferspeicher 51 übertragen. Zu dieser Zeit beträgt die zu übertragende Bilddatenmenge 384 Pixel.

Anschließend wird zur Bewegungsvoraussage die Helligkeitskomponente eines Referenzbildes (reproduziertes Bild) im Bereich der Bewegungssuche durch den Pixeldatenbus PB aus dem Bildpufferspeicher 51 in eine Einheit zur Bewegungsvoraussage 46 übertragen. Zu dieser Zeit beträgt die auf ein Einzelbild des Referenzbilds bezogene zu übertragende Bilddatenmenge 768 Pixel. Daher müssen bei einer Zwei-Weg-Voraussage 1536 Pixel, das heißt zwei Bildebenen, übertragen werden.

Dann wird zur Bewegungsvoraussage die Helligkeitskomponente des zu codierenden Bildes durch den Pixeldatenbus PB aus dem Bildpufferspeicher 51 in die Steuereinheit 2 übertragen.

Anschließend wird die Helligkeitskomponente aus der Steuereinheit 2 durch den lokalen Bus LB3 in die Einheit zur Bewegungsvoraussage 46 übertragen. Zu dieser Zeit beträgt die übertragene Bilddatenmenge 256 Pixel.

Die Bewegungsvoraussage wird beendet, und es ist eine Zwischenbildvoraussage gemäß der festgelegten Codierweise unter Verwendung der Ergebnisse auszuführen, wobei die Daten des vorausgesagten Bildes (reproduzierten Bildes) zum Codieren aus dem Bildpufferspeicher 51 durch den Pixeldatenbus PB in die Pixelverarbeitungseinheit 3 übertragen werden. Zu dieser Zeit beträgt die zu übertragende Bilddatenmenge 451 Pixel für eine Ebene des vorausgesagten Bildes. Daher müssen für die Zwei-Weg-Voraussage 902 Pixel für zwei Bildebenen übertragen werden.

Danach werden die zu codierenden Bilddaten aus dem Bildpufferspeicher 51 durch den Pixeldatenbus PB in die Pixelverarbeitungseinheit 3 übertragen. Zu dieser Zeit beträgt die übertragene Bilddatenmenge 384 Pixel.

Nach Beendigung des Codierens werden die Daten des zu reproduzierenden Bildes bezüglich des Codebildes eines Innenbildes und des Bildes der Ein-Weg-Voraussage durch den Pixeldatenbus PB aus der Pixelverarbeitungseinheit 3 in den Bildpufferspeicher 51 übertragen. Zu dieser Zeit beträgt die übertragene Bilddatenmenge 384 Pixel.

Die vorstehend beschriebene Datenübertragungsmenge, welche Menge 3462 Pixel beträgt, wird zur Zeit der Zwei-Weg-Voraussage maximal, bei welcher die gesamte Datenübertragung ausgeführt wird, abgesehen von der Übertragung der zu reproduzierenden Bilddaten.

Wenn nun der Bereich der Bewegungssuche vergrößert wird, um die Bildqualität zu verbessern, dann nimmt die Bilddaten-Übertragungsmenge der Helligkeitskomponente des Referenzbildes im Bereich der Bewegungssuche zu. Wenn beispielsweise der Bereich der Bewegungssuche auf  $\pm 32$  Pixel vergrößert wird, dann wird es erforderlich, 1280 Pixel zur Zeit der Ein-Weg-Voraussage und 2560 Pixel zur Zeit der Zwei-Weg-Voraussage zu übertragen.

Wie für Digitalbilddaten existieren unterschiedliche Datenformate, welche von der Datenmenge der Farbdifferenzkomponente der Bilddaten abhängig sind. Es gibt nämlich 4 : 4 : 4 Formate, bei welchen die Menge der Originalbilddaten aufrechterhalten wird, ein Format 4 : 2 : 2, bei welchem die Farbdifferenzkomponente auf die Hälfte verkleinert ist, und ein Format 4 : 2 : 0, bei welchem die Farbdifferenzkomponente auf ein Viertel verkleinert ist. Wenn daher das zu bearbeitende Bildformat verändert wird, dann nimmt die Bilddaten-Übertragungsmenge zu, abgesehen von der Bilddaten-Übertragungsmenge bezüglich der Bewegungsvoraussage. Wenn zum Beispiel das Format 4 : 4 : 4 verwendet wird, dann wird die bei der Übertragung der Originalbilddaten, die bei der Übertragung der zu codierenden Bilddaten zur Pixelverarbeitungseinheit und die bei der Übertragung der zu reproduzierenden Bilddaten bearbeitete Datenmenge 768 Pixel betragen, und wie für die Übertragung der Daten zur Bildvoraussage, müssen, bezogen auf eine Bildebene des vorausgesagten Referenzbildes, nahezu zweimal so viele Pixel, das heißt 867 Pixel, übertragen werden.

Wenn, wie vorstehend beschrieben, die Anzahl der Pixel des zu verarbeitenden Bildes zunimmt, wie im Fall eines hochgradig festgelegten Bildes, dann nimmt die zur Verarbeitung eines Makroblocks erforderliche Zeit zu. Wenn die Systemuhr konstant läuft, dann ist die Datenübertragungsmenge, die in einem Zyklus der Systemuhr bearbeitet werden kann, durch die Leistungsfähigkeit der Hardware begrenzt, da die Datenübertragungsmenge zur Verarbeitung eines Makroblocks zunimmt.

Daher nimmt beim Übertragen der vorstehend beschriebenen Daten die Anzahl der Zyklen der Systemuhr zu.

Wenn, wie vorstehend beschrieben, die Datenübertragungsmenge sehr groß ist und ständig zunimmt, da eine größere Anzahl von Pixeln und eine höhere Bildqualität verlangt werden, dann begrenzt der Datenübertragungsprozeß die Verarbeitungsgeschwindigkeit der ganzen Vorrichtung, wenn die Pipelineverarbeitung unter Verwendung einer Makroblockcodierung als eine Einheit ausgeführt wird, da die Anzahl von Zyklen, die zur Datenübertragung zum Ausführen des Codierens erforderlich ist, größer als die Anzahl derjenigen Zyklen ist, die für den Codierbetrieb an sich erforderlich ist. Im folgenden wird eine Vorrichtung zur Bildkompression/-expansion beschrieben werden, welche die Effizienz bei der Datenübertragung verbessert und deren Verarbeitungsgeschwindigkeit insgesamt nicht durch die Datenübertragungsgeschwindigkeit begrenzt ist.

#### Die siebente Ausführungsform

Fig. 83 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion gemäß einer siebenten Ausführungsform der vorliegenden Erfindung dar. In Fig. 83 sind die Abschnitte, welche denen der in Fig. 8 gezeigten sechsten Ausführungsform entsprechen, mit den gleichen Bezugszeichen bezeichnet, und die Beschreibung derselben wird nicht wiederholt.

Unter Bezugnahme auf Fig. 83 weist ein Pixeldatenbus PB eine Datenbreite von 32 Bit auf und ist zur Datenübertragung zwischen einer Steuereinheit 23, einer Pixelverarbeitungseinheit 33 und einem Bildpufferspeicher 51 vorgesehen.

Ein lokaler Datenbus LDB weist eine Datenbreite von 32 Bit auf und ist zur Datenübertragung zwischen der

Pixelverarbeitungseinheit 33, einer Einheit zur Bewegungsvoraussage 47 und einem lokalen Speicher 56 vorgesehen.

Der lokale Speicher 56 ist zum zeitweiligen Speichern einer aus der Pixelverarbeitungseinheit 33 übertragenen Helligkeitskomponente von reproduzierten Bilddaten vorgesehen, die zur Bewegungsvoraussage verwendet werden. Ein Speicher mit größerer Speicherkapazität, wie beispielsweise ein SRAM, ein DRAM, ein synchroner DRAM und ein Cache-DRAM, wird als lokaler Speicher 56 verwendet.

Ein in der Steuereinheit 23 enthaltener Mikroprozessor (nicht dargestellt) zur Gesamtsteuerung steuert die Bilddatenübertragung zwischen der Steuereinheit 23, dem Bildpufferspeicher 51 und der Pixelverarbeitungseinheit 33 und steuert die Bilddatenübertragung zwischen der Pixelverarbeitungseinheit 33, der Einheit zur Bewegungsvoraussage 47 und dem lokalen Speicher 56.

Die Pixelverarbeitungseinheit 33 ist zusätzlich zu einem Dateneingangs-/Datenausgangsport P1, das mit dem Pixeldatenbus PB verbunden ist, mit einem neuen Datenausgangsport P2 für Bilddaten versehen. Mit dem Datenausgangsport P2 sind die Einheit zur Bewegungsvoraussage 47 und der neu vorgesehene lokale Speicher 56 durch einen lokalen Datenbus LDB verbunden. Das Datenausgangsport P2 der Pixelverarbeitungseinheit 33, die Einheit zur Bewegungsvoraussage 47 und der lokale Speicher 56 werden durch die Steuereinheit 23 gesteuert und führen die Bilddatenübertragung durch einen lokalen Datenbus 61 aus. Die Steuereinheit 23 steuert ferner das Timing von jeder Übertragung.

Die vom lokalen Datenbus LDB geführten Bilddaten stellen die Referenzdaten zur Bewegungsvoraussage dar, die aus den Bilddaten gewonnen werden, welche durch die Pixelverarbeitungseinheit 33 aus dem Bildpufferspeicher 51 durch den Pixeldatenbus PB übertragen werden (Die Daten enthalten wenigstens eine Information, die bei den Bilddaten auf Helligkeitsdaten hinweist, welche nachstehend als Helligkeitskomponente bezeichnet wird.). Die Referenzdaten werden durch den Pixeldatenbus PB aus dem Dateneingangs-/Datenausgangsport P1 zum Bildpufferspeicher 51 übertragen und darin gespeichert.

Ein zugeordneter lokaler Bus LB4 mit einer Breite von 8 Bit ist zwischen der Einheit zur Bewegungsvoraussage 47 und der Steuereinheit 23 vorgesehen. Die Bilddaten, welche Gegenstand der Bewegungsermittlung sind, werden aus der Steuereinheit 23 zur Einheit zur Bewegungsvoraussage 47 übertragen. Die das Ergebnis der Bewegungsermittlung angegebenden Bewegungsvektor-Daten werden aus der Einheit zur Bewegungsvoraussage 47 zur Steuereinheit 23 übertragen. Die Steuereinheit 23, der Bildpufferspeicher 51 und die Pixelverarbeitungseinheit 33, welche mit dem Pixeldatenbus PB verbunden sind, bewirken auf Grundlage der Bewegungsvektordaten die Erzeugung eines vorausgesagten Bildes und das Codieren des vorausgesagten Bildes. Die Pixelverarbeitungseinheit 33 stellt die codierten Bilddaten einem Codebus CB zur Verfügung.

Die Bilddatenübertragung bei der Bildcodierung und die Bilddatenübertragungsmenge beim Verarbeiten eines Makroblocks der Bilddaten werden unter Bezugnahme auf Fig. 83 beschrieben werden. Dabei wird angenommen, daß die zu codierenden Bilddaten bereits im Bildpufferspeicher 51 gespeichert worden ist und daß die Helligkeitskomponente des für die Bewegungsermittlung erforderlichen Referenzbildes (des reproduzierten Bildes) bereits im lokalen Speicher 56 gespeichert worden ist. Die Übertragungsmenge wird unter der Voraussetzung berechnet, daß das Bildformat 4 : 2 : 0 beträgt, der Bereich der Bewegungsermittlung  $\pm 16$  Pixel umfaßt und die Struktur eine Bildstruktur aufweist.

Zuerst werden die Originalbilddaten durch den Pixeldatenbus PB aus der Steuereinheit 23 in den Bildpufferspeicher 51 übertragen. Zu dieser Zeit beträgt die Menge der übertragenen Bilddaten 384 Pixel.

Dann wird zur Bewegungsvoraussage die Helligkeitskomponente des Bildes, welche Gegenstand der Bewegungsvoraussage ist, durch den Pixeldatenbus PB aus dem Bildpufferspeicher 51 in die Steuereinheit 23 übertragen. Die Helligkeitskomponente wird ferner durch einen lokalen Bus LB4 in die Einheit zur Bewegungsermittlung 47 übertragen. Zu dieser Zeit beträgt die Menge der übertragenen Bilddaten 256 Pixel.

Wenn die Bewegungsvoraussage abgeschlossen ist und die auf den Ergebnissen der Bewegungsvoraussage beruhende Zwischenbildvoraussage gemäß der bestimmten Codierweise auszuführen ist, dann wird zum Codieren das vorausgesagte (zu reproduzierende) Bild durch den Pixeldatenbus PB aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 33 übertragen. Zu dieser Zeit beträgt die auf eine Ebene des vorausgesagten Bildes bezogene Menge der übertragenen Bilddaten 451 Pixel. Daher müssen zur Zwei-Weg-Voraussage 902 Pixel übertragen werden.

Anschließend werden die zu codierenden Bilddaten durch den Pixeldatenbus PB aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 33 übertragen. Zu dieser Zeit beträgt die Menge der übertragenen Bilddaten 384 Pixel.

Parallel zu den vorstehend genannten vier Bilddatenübertragungen wird zur Bewegungsvoraussage die Helligkeitskomponente des Bildes im Bereich der Bewegungssuche durch den lokalen Datenbus LDB aus dem lokalen Speicher 56 zur Einheit zur Bewegungsermittlung 47 übertragen. Zu dieser Zeit beträgt die auf eine Ebene des Referenzbildes bezogene Menge der übertragenen Bilddaten 768 Pixel. Daher müssen zur Zwei-Weg-Voraussage, den zwei Ebenen entsprechend, 1536 Pixel übertragen werden.

Nach Abschluß des Codierens bezüglich des Bildes für die Innenbildcodierung und des Bildes für die Ein-Weg-Voraussage werden die zu reproduzierenden Bilddaten durch den Pixeldatenbus PB aus der Pixelverarbeitungseinheit 33 in den Bildpufferspeicher 51 übertragen. Gleichzeitig werden die Helligkeitskomponenten des zu reproduzierenden Bildes durch den lokalen Datenbus LDB in den lokalen Speicher 56 übertragen.

Zu dieser Zeit stellt die Steuereinheit 23 ein Ausgangssteuersignal zum Steuern des mit dem Pixeldatenbus PB verbundenen Ports P1 in Synchronisation mit dem Timing des Ausgangs der Bilddaten zur Verfügung, welche durch die Pixelverarbeitungseinheit 33 aus einem internen Reproduktionsbild-Ausgabespeicher (zum Beispiel dem in Fig. 19 dargestellten Bildspeicher 335, der nachstehend beschrieben werden wird) zu reproduzieren sind. Während die Helligkeitsdaten aus dem Reproduktionsbild-Ausgabespeicher ausgegeben werden, stellt die Steuereinheit 23 gleichzeitig ein Ausgangssteuersignal zum Steuern des mit dem lokalen Datenbus LDB verbun-

denen Ports P2 zur Verfügung.

In der Pixelverarbeitungseinheit 33 werden die Bilddaten aus dem Reproduktionsbild-Ausgabespeicher gemäß einer Anweisung aus der Steuereinheit 23 oder gemäß einem Ablaufprogramm ausgegeben, das in einem internen Speicher gespeichert ist (zum Beispiel einem internen Speicher in einer in Fig. 19 dargestellten Steuereinrichtung, welche nachstehend beschrieben werden wird). Zu dieser Zeit stellt das mit dem Pixeldatenbus PB verbundene Port P1 in Reaktion auf ein aus der Steuereinheit 23 bereitgestelltes Ausgangssteuersignal Bilddaten von einem Makroblock zur Verfügung. Gleichzeitig stellt das mit dem lokalen Datenbus LDB verbundene Port P2 in Reaktion auf das aus der Steuereinheit 23 bereitgestellte Ausgangssteuersignal nur die Helligkeitskomponente der Bilddaten von einem Makroblock zur Verfügung.

Bei der vorstehend beschriebenen Übertragung der zu reproduzierenden Bilder beträgt die Menge der übertragene Bilddaten auf dem Pixeldatenbus PB 384 Pixel, auf dem lokalen Datenbus LDB hingegen 256 Pixel.

Wie vorstehend beschrieben, ist die Pixelverarbeitungseinheit 33 zum Erzeugen eines zu reproduzierenden Bildes derart ausgelegt, daß sie zu zwei Datenbussen eine Datenausgabefunktion aufweist und das zu reproduzierende Bild zum lokalen Datenbus LDB in Synchronisation mit der Ausgabe des zu reproduzierenden Bildes an den Pixeldatenbus PB ausgibt. Daher kann der Reproduktionsbild-Ausgabespeicher (nicht dargestellt) in der Pixelverarbeitungseinheit 33 gemeinsam gesteuert werden, und die Daten können zu jedem der mit dem internen Speicherausgang verbundenen Ports durch nur ein Bit des Ausgangssteuersignals übertragen werden. Im Ergebnis wird die Datenübertragungssteuerung hinsichtlich der Steuereinheit 23 erleichtert und, von der Übertragung des Reproduktionsbildes abgesehen, werden die Busse durch die Steuereinheit 23 aufgeteilt, wodurch eine parallele Datenübertragung ermöglicht wird.

Die Datenübertragungsmenge zu dieser Zeit wird zur Zeit der Zwei-Weg-Voraussage maximal, nämlich 1926 Pixel auf dem Pixeldatenbus PB und 1536 Pixel auf dem lokalen Datenbus LDB. Durch Verwenden der aufgeteilten zwei Datenbusse kann daher die durch einen Bus übertragene Datenmenge im Vergleich zur sechsten Ausführungsform verkleinert werden, wobei die tatsächliche Datenübertragungsgeschwindigkeit verbessert wird und die Verarbeitungsgeschwindigkeit in der Vorrichtung zur Bildkompression/-expansion vergrößert wird.

#### Die achte Ausführungsform

Fig. 84 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 108 dar, welche eine achte Ausführungsform der vorliegenden Erfindung zeigt. In Fig. 84 sind die Abschnitte, welche denen der in Fig. 83 gezeigten Vorrichtung zur Bildkompression/-expansion entsprechen, mit den gleichen Bezugszeichen versehen, und die Beschreibung derselben wird nicht wiederholt.

Eine Pixelverarbeitungseinheit 34 umfaßt zwei neu vorgesehene Bilddaten-Ausgangsports P4 und PS. Drei Datenbusse, das heißt ein Pixeldatenbus PB und zwei lokale Datenbusse LDB1 und LDB2, sind als Bilddatenbusse vorgesehen. Der Ausgangsteuerbetrieb der entsprechenden Ports P3 bis PS wird durch ein Ausgangssteuersignal bewirkt, das aus einer Systemsteuereinrichtung (zum Beispiel eine in Fig. 13 dargestellte Systemsteuereinrichtung 114, welche nachstehend beschrieben werden wird) bereitgestellt wird, welche in einem Prozessor zur Pipelinesteuerung (beispielsweise einem in Fig. 9 dargestellten Prozessor zur Gesamtsteuerung 11, der nachstehend beschrieben werden wird) in einer Steuereinheit 24 enthalten ist.

Wenn die Funktion der Vorrichtung erweitert und der Bereich der Bewegungsermittlung auf  $\pm 64$  Pixel vergrößert wird, dann müssen selbst zur Zeit der Ein-Weg-Voraussage Bilddaten mit 2304 Pixeln übertragen werden. Im folgenden wird eine Beschreibung unter der Voraussetzung vorgenommen, daß der Bereich der Bewegungsermittlung auf  $\pm 64$  Pixel vergrößert wird und die Ein-Weg-Voraussage ausgeführt wird. Bei der in Fig. 8 dargestellten sechsten Ausführungsform ist die Bilddatenübertragungsmenge auf 4230 Pixel vergrößert, da der Bereich der Bewegungsermittlung vergrößert ist. Das Verfahren bei der achten Ausführungsform ist folgendermaßen. In der achten Ausführungsform wird zur Bewegungsvoraussage die Helligkeitskomponente des Bildes im Bereich der Bewegungssuche in zwei Komponenten aufgeteilt und unter Verwendung von zwei Datenbussen übertragen. Insbesondere werden 1280 Pixel durch einen lokalen Datenbus LDB1 aus einem lokalen Speicher 56a zu einer Voraussageeinheit 47a übertragen, während 1280 Pixel durch einen lokalen Datenbus LDB2 aus einem lokalen Speicher 56b zu einer Einheit zur Bewegungsvoraussage 47b übertragen werden. Zu dieser Zeit müssen die Daten am Verbindungsabschnitt der Daten zu beiden Datenbussen bDB1 und LDB 2 übertragen werden. Daher würde die durch zwei Datenbusse übertragene Datenmenge insgesamt 2560 Pixel betragen, obwohl die durch einen Datenbus übertragene Datenmenge 2304 Pixel beträgt. Mit anderen Worten, die Datenmenge wird durch die Pixeldaten von einem Makroblock (256 Pixel) vergrößert. Das dient dazu, ein Erkennen der Verbindung zwischen entsprechenden Stücken der aufgeteilten Referenzdaten zu ermöglichen.

Wenn in diesem Fall das zu reproduzierende Bild übertragen wird, werden die Reproduktionsbilddaten durch den Pixeldatenbus PB aus der Pixelverarbeitungseinheit 34 in den Bildpufferspeicher 51 übertragen, und gleichzeitig wird das Reproduktionsbild durch die lokalen Datenbusse LDB1 und LDB2 in die lokalen Speicher 56a und 56b übertragen.

Die Datenübertragungsmenge zu dieser Zeit beträgt 1926 Pixel auf dem Pixeldatenbus PB und 1280 Pixel auf jedem der lokalen Datenbusse LDB1 und LDB2. Insbesondere kann die Menge der über einen Bus übertragenen Daten verkleinert werden, da der Bilddatenbus in drei Busse aufgeteilt ist, wobei die tatsächliche Datenübertragungsgeschwindigkeit auf über das Doppelte von der der sechsten Ausführungsform vergrößert wird und die siebente Verarbeitung der Vorrichtung zur Bildkompression/-expansion verbessert werden kann.



## Die neunte Ausführungsform

Fig. 85 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion gemäß einer neunten Ausführungsform der vorliegenden Erfindung dar. In Fig. 85 sind die Abschnitte, welche denen der in Fig. 84 gezeigten Vorrichtung zur Bildkompression/-expansion entsprechen, mit den gleichen Bezugszeichen bezeichnet, und die Beschreibung derselben wird nicht wiederholt.

Unter Bezugnahme auf Fig. 85 verwendet die neunte Ausführungsform einen gemeinsamen Adreßbus AB für lokale Speicher 56c und 56d. Der Adreßbus AB ist mit einer Steuereinheit 25 verbunden, und die Adreßsignale werden aus der Steuereinheit 25 ausgegeben.

Wenn bei der Zwei-Weg-Voraussage als erweiterte Funktion der Vorrichtung zur Bildkompression/-expansion der Bereich der Bewegungssuche auf  $\pm 32$  Pixel gesetzt wird, dann wird es erforderlich, 1280 Pixel pro 1 Ebene des vorausgesagten Bildes zu übertragen. Nachstehend wird die Bilddatenübertragung beschrieben werden, wobei vorausgesetzt wird, daß die Bewegung im Bereich von  $\pm 32$  Pixeln gesucht wird und daß eine Zwei-Weg-Voraussage ausgeführt wird.

Zuerst wird die Datenübertragung im Fall von Bildern für eine Innenbildcodierung beschrieben werden. Wenn beispielsweise im lokalen Speicher 56d gespeicherte zu reproduzierende Bilddaten im Vergleich zu den im lokalen Speicher 56c gespeicherten Bilddaten zur Reproduktion neuer (zeitlich später) sind oder wenn die Ausgangsbilddaten zu codieren sind (die Daten sind in keinem der lokalen Speicher 56c und 56d gespeichert), dann wird die Datenübertragung in gleichartiger Weise wie in der siebenten Ausführungsform ausgeführt, wobei die Steuereinheit 25, der Bildpufferspeicher 51, die Pixelverarbeitungseinheit 34, die Einheit zur Bewegungsvoraussage 47a, der lokale Speicher 56c, der Pixeldatenbus PB und der lokale Datenbus LDB1 verwendet werden. Das Helligkeitssignal des zu reproduzierenden Bildes ist im lokalen Speicher 56c gespeichert, um es für Referenzbilddaten bei der Bewegungsvoraussage zu verwenden. Zu dieser Zeit steuert die Steuereinheit 25 ein Schreiben von Daten durch Wählen des lokalen Speichers 56c durch ein Chipauswahlsignal und stellt ein Ausgangssteuersignal bereit, um das Port P4 auf der Seite des lokalen Datenbusses LDB1 der Pixelverarbeitungseinheit 34 auszugeben.

Die Datenübertragung des Bildes für die Ein-Weg-Voraussage wird beschrieben werden. Wenn beispielsweise die im lokalen Speicher 56c gespeicherten Reproduktionsbilddaten im Vergleich zu den im lokalen Speicher 56d gespeicherten Reproduktionsbilddaten neuer (zeitlich später) sind, dann werden die im lokalen Speicher 56c gespeicherten Reproduktionsbilddaten als Referenzbilddaten für eine Vorwärtsvoraussage verwendet und die Bilddaten im Bereich der Bewegungsvoraussagesuche unter Verwendung der Einheit zur Bewegungsvoraussage 47a, des lokalen Speichers 56c und des lokalen Datenbusses LDB1 übertragen. Nur wenn das Helligkeitssignal des zu reproduzierenden Bildes im Bereich der Bewegungsvoraussagesuche zu übertragen ist, steuert zu dieser Zeit die Steuereinheit 25 die Datenausgabe durch Wählen des lokalen Speichers 56c durch das Chipauswahlsignal und legt ein Eingangssteuersignal des Eingangsports an die Seite des lokalen Datenbusses der Einheit zur Bewegungsvoraussage 47a.

Bei einer von der vorstehenden abweichenden Bildübertragung (zum Beispiel Originalbildübertragung, Übertragung eines zu codierenden Bildes zur Bewegungsvoraussage, Übertragung von Bildern zur Vorwärtsvoraussage, Übertragung von zu codierenden Bildern und Übertragung von zu reproduzierenden Bildern) werden die Daten in gleichartiger Weise wie in der siebenten Ausführungsform durch Verwenden der Steuereinheit 25, des Bildpufferspeichers 51, der Pixelverarbeitungseinheit 34, des lokalen Speichers 56d, des Pixeldatenbusses PB und des lokalen Datenbusses LDB2 übertragen. Zu dieser Zeit wird die Datenübertragungsmenge über den Pixeldatenbus PD zur Zeit der Zwei-Weg-Voraussage maximal 1926 Pixel betragen, und die Datenübertragungsmenge über die lokalen Datenbusse LDB1 und LDB2 wird jeweils maximal 1280 Pixel betragen. Dabei wird das Helligkeitssignal des zu reproduzierenden Bildes im lokalen Speicher 56d als Referenzbilddatenangabe zur Bewegungsvoraussage gespeichert. Zu dieser Zeit steuert die Steuereinheit 25 das Schreiben von Daten durch Wählen des lokalen Speichers 56d auf Grundlage des Chipauswahlsignals nur dann, wenn das Helligkeitssignal des zu reproduzierenden Bildes übertragen wird, und legt ein Ausgangssignal an das Ausgangsportal PS auf der Seite des lokalen Datenbusses LDB2 der Pixelverarbeitungseinheit 34. Dabei wird das Adressieren des lokalen Speichers 56d in gleichartiger Weise wie das Adressieren des lokalen Speichers 56c ausgeführt.

Die Datenübertragung für das Bild bei der Zwei-Weg-Voraussage wird beschrieben werden. Wenn beispielsweise die im lokalen Speicher 56d gespeicherten Daten des zu reproduzierenden Bildes im Vergleich zu den im lokalen Speicher 56c gespeicherten Daten des zu reproduzierenden Bildes neuer (zeitlich später) sind, dann werden die im lokalen Speicher 56c gespeicherten Bilddaten für die Reproduktion als Referenzbilddaten für eine Vorwärtsvoraussage verwendet, und die Bildübertragung im Bereich der Bewegungsvoraussagesuche wird durch Verwenden der Einheit zur Bewegungsvoraussage 47a, des lokalen Speichers 56c und des lokalen Datenbusses LDB1 durchgeführt. Gleichzeitig wird das im lokalen Speicher 56b gespeicherte zu reproduzierende Bild als Referenzbild zur Rückwärtsvoraussage verwendet und die Übertragung des Bildes im Bereich der Bewegungsvoraussagesuche durch Verwenden der Einheit zur Bewegungsvoraussage 47b, des lokalen Speichers 56d und des lokalen Datenbusses LDB2 bewirkt. Zu dieser Zeit wählt die Steuereinheit 25 die lokalen Speicher 56c und 56d durch das Chipauswahlsignal nur dann aus, wenn das Helligkeitssignal des zu reproduzierenden Bildes im Bereich der Bewegungsvoraussagesuche zu übertragen ist, so daß die gemeinsame Datenausgangssteuerung durch Anlegen des gleichen Adreßsignals unter Verwendung des gemeinsamen Adreßbusses AB ausgeführt wird. Die Steuereinheit 25 legt ferner ein Eingangssteuersignal des Eingangsports an die Seite der lokalen Datenbusse LDB1 und LDB2 der entsprechenden Einheiten zur Bewegungsvoraussage 47a und 47b.

Bei von den vorstehenden abweichenden Bildübertragungen (zum Beispiel: Originalbildübertragung, Übertragung eines zu codierenden Bildes zur Bewegungsvoraussage, Übertragung von Bildern zur Vorwärtsvoraussage, Übertragung von Bildern zur Rückwärtsvoraussage und Übertragung von zu codierenden Bildern) werden



die Bilddaten in gleichartiger Weise wie in der siebenten Ausführungsform durch Verwenden der Steuereinheit 25, des Bildpufferspeichers 51, der Pixelverarbeitungseinheit 34 und des Pixeldatenbusses PB übertragen.

Nun weisen der Bereich der Vorwärtsbewegungssuche und der Bereich der Rückwärtsbewegungssuche bezüglich eines Makroblocks, der Gegenstand der Bewegungsvoraussage ist, den gleichen Bereich auf dem tatsächlichen Bildschirm auf. Daher wird es durch gemeinsames Adressieren der lokalen Speicher 56c und 56d möglich, die Daten im Bereich der Voraussagesuche durch Verwenden einer gemeinsamen Adresse in beide Richtungen zu übertragen. Abgesehen vom Chipauswahlsignal der lokalen Speicher 56c und 56d, kann insbesondere die Hardware bezüglich der Speichersteuerung und der Adreßerzeugung durch die lokalen Speicher 56c und 56d aufgeteilt werden. Daher kann im Vergleich zu einem Beispiel, bei dem Adreßgeneratoren für entsprechende Speicher vorgesehen sind, die Hardwaremenge verringert werden, wohingegen die Datenübertragungsmenge nicht erhöht wird.

#### Die zehnte Ausführungsform

Fig. 86 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion dar, welche eine zehnte Ausführungsform der vorliegenden Erfindung zeigt. In Fig. 86 sind die Abschnitte, welche denen der in Fig. 85 gezeigten Vorrichtung zur Bildkompression/-expansion entsprechen, mit den gleichen Bezugszeichen bezeichnet, und die Beschreibung derselben wird nicht wiederholt.

Unter Bezugnahme auf Fig. 86 ist in der zehnten Ausführungsform die in Fig. 85 dargestellte Pixelverarbeitungseinheit 34 in zwei Pixelverarbeitungseinheiten 34a und 34b unterteilt.

Wenn die Bewegungssuche im Bereich von  $\pm 32$  Pixeln für eine Zwei-Weg-Voraussage als erweiterte Funktion der Vorrichtung zur Bildkompression/-expansion auszuführen ist, dann ist es erforderlich, bezüglich einer Ebene des vorausgesagten Bildes 1280 Pixel zu übertragen. Nachstehend wird eine Bilddatenübertragung unter der Annahme beschrieben werden, daß die Bewegungssuche im Bereich von  $\pm 32$  Pixeln ausgeführt wird und daß die Zwei-Weg-Voraussage vollzogen wird.

Zuerst wird die Datenübertragung des Bildes für die Innenbildcodierung beschrieben werden. Wenn beispielsweise die in einem lokalen Speicher 56b gespeicherten zu reproduzierenden Daten des Bildes neuer (zeitlich später) als die in einem lokalen Speicher 56c gespeicherten Daten des zu reproduzierenden Bildes sind oder wenn die Ausgangsbilddaten zu codieren sind (die Daten sind in keinem der lokalen Speicher 56c oder 56d gespeichert), dann wird die Datenübertragung in gleichartiger Weise wie in der siebenten Ausführungsform ausgeführt durch Verwenden einer Steuereinheit 26, eines Bildpufferspeichers 51, einer Pixelverarbeitungseinheit, einer Einheit zur Bewegungsvoraussage 47a, des lokalen Speichers 56c, eines Pixeldatenbusses PB und eines lokalen Datenbusses LDB1. Das Helligkeitssignal des zu reproduzierenden Bildes ist im lokalen Speicher 56c gespeichert, um es für Referenzbilddaten zur Bewegungsvoraussage zu verwenden. Zu dieser Zeit steuert eine Steuereinheit 26 nur dann ein Schreiben von Daten durch Wählen des lokalen Speichers 56c durch ein Chipauswahlsignal, wenn das Helligkeitssignal des zu reproduzierenden Bildes übertragen wird. Die Steuereinheit 36 stellt ein Ausgangssteuersignal an einem Port P2a auf der Seite des lokalen Datenbusses LDB1 der Pixelverarbeitungseinheit 34a bereit.

Die Bilddatenübertragung für eine Ein-Weg-Voraussage wird beschrieben werden. Wenn beispielsweise die im lokalen Speicher 56c gespeicherten Daten des zu reproduzierenden Bildes im Vergleich zu den im lokalen Speicher 56d gespeicherten Daten des zu reproduzierenden Bildes neuer (zeitlich später) sind, dann werden die im lokalen Speicher 56c gespeicherten Bilddaten für die Reproduktion als Referenzbilddaten für eine Vorwärts-voraussage verwendet und wird die Bildübertragung im Bereich der Bewegungsvoraussagesuche unter Verwendung der Einheit zur Bewegungsvoraussage 47a, des lokalen Speichers 56c und des lokalen Datenbusses LDB1 ausgeführt. Zu dieser Zeit steuert die Steuereinheit 26 die Datenausgabe durch Wählen des lokalen Speichers 56c auf Grundlage des Chipauswahlsignals nur dann, wenn das Helligkeitssignal des zu reproduzierenden Bildes im Bereich der Bewegungsvoraussagesuche übertragen wird. Die Steuereinheit 26 legt ein Eingangssteuersignal des Eingangsports an die Seite des lokalen Datenbusses LDB1 der Einheit zur Bewegungsvoraussage 47a.

Bei einer von der vorstehenden abweichenden Bildübertragung (Übertragung des Originalbildes, Übertragung eines zu codierenden Bildes zur Bewegungsvoraussage, Übertragung eines Bildes zur Vorwärts-voraussage, Übertragung eines zu codierenden Bildes und Übertragung eines zu reproduzierenden Bildes) werden die Daten in gleichartiger Weise wie in der siebenten Ausführungsform durch Verwenden der Steuereinheit 26, des Bildpufferspeichers 51, der Pixelverarbeitungseinheit 34b, des lokalen Speichers 56d, des Pixeldatenbusses PB und des lokalen Datenbusses LDB2 übertragen. Das Helligkeitssignal des zu reproduzierenden Bildes wird im lokalen Speicher 56d gespeichert, um es als Referenzbild zur Bewegungsvoraussage zu verwenden. Zu dieser Zeit steuert die Steuereinheit 26 das Schreiben von Daten durch Wählen des lokalen Speichers 56c auf Grundlage des Chipauswahlsignals nur dann, wenn das Helligkeitssignal des zu reproduzierenden Bildes übertragen wird. Die Steuereinheit legt ein Ausgangssignal an ein Ausgangs-port P2b auf der Seite des lokalen Datenbusses LDB2 der Pixelverarbeitungseinheit 34b. Dabei wird das Adressieren des lokalen Speichers 56b in gleichartiger Weise wie das Adressieren des lokalen Speichers 56c bewirkt.

Die Datenübertragung für das Bild bei der Zwei-Weg-Voraussage wird beschrieben werden. Wenn beispielsweise die im lokalen Speicher 56b gespeicherten Daten des zu reproduzierenden Bildes im Vergleich zu den im lokalen Speicher 56c gespeicherten Daten des zu reproduzierenden Bildes neuer (zeitlich später) sind, dann werden die im lokalen Speicher 56c gespeicherten Daten des zu reproduzierenden Bildes als Referenzbilddaten für eine Vorwärts-voraussage verwendet, und das Bild im Bereich der Bewegungsvoraussagesuche wird durch Verwenden der Einheit zur Bewegungsvoraussage 47a, des lokalen Speichers 56c und des lokalen Datenbusses LDB1 übertragen. Zu dieser Zeit werden die Daten des im lokalen Speicher 56d gespeicherten zu reproduzierenden Bildes als Referenzbilddaten zur Rückwärts-voraussage verwendet, und das Bild im Bereich der Bewegungs-

vorausgesagte wird durch Verwenden der Einheit zur Bewegungsvoraussage 47b, des lokalen Speichers 56b und des lokalen Datenbusses LDB2 übertragen. Zu dieser Zeit wählt die Steuereinheit 26 die lokalen Speicher 56c und 56d durch das Chipauswahlsignal nur dann aus, wenn das Helligkeitssignal des zu reproduzierenden Bildes im Bereich der Bewegungsvorausage übertragen wird, so daß die Datenausgabe durch Anlegen der gleichen Adresse unter Verwendung eines gemeinsamen Adreßbusses AB gemeinsam gesteuert wird. Die Steuereinheit 26 legt ferner ein Eingangssteuersignal eines Eingangsabschnitts an die Seite der lokalen Datenbusse LDB1 und LDB2 der Einheiten zur Bewegungsvoraussage 47a und 47b.

Bei einer von der vorstehenden abweichenden Bildübertragung (Übertragung eines Originalbildes, Übertragung eines zu codierenden Bildes zur Bewegungsvoraussage, Übertragung eines Bildes zur Vorwärtsvoraussage, Übertragung eines Bildes zur Rückwärtsvoraussage und Übertragung eines zu codierenden Bildes) werden die Bilddaten wie in der in Fig. 7 dargestellten Art und Weise durch Verwenden der Steuereinheit 26, des Bildpufferspeichers 51, der Pixelverarbeitungseinheit 34a oder 34b und des Pixeldatenbusses PB übertragen. Zu dieser Zeit wird das Bild Makroblock für Makroblock durch abwechselndes Verwenden der beiden Pixelverarbeitungseinheiten 34a und 34b übertragen, und in der Verarbeitungseinheit 34a oder 34b wird das Codieren von zwei Makroblöcken ausgeführt, wobei die Zeit zur Verarbeitung von 2 Makroblöcken verwendet wird.

Wenn eine Zwischenbild-Vorausagecodierung auszuführen ist, dann ist die Menge an Operationen bei der Zwei-Weg-Vorausage im allgemeinen sehr groß. In der vorliegenden Ausführungsform werden zur Zeit der Zwei-Weg-Vorausage die Bilddaten Makroblock für Makroblock abwechselnd in die beiden Verarbeitungseinheiten 34a und 34b übertragen, so daß der einen Pixelverarbeitungseinheit 34a oder 34b im wesentlichen die Operationszeit von zwei Makroblöcken gewährt wird. Daher kann im Vergleich zur neunten Ausführungsform die tatsächliche Operationszeit der Pixelverarbeitungseinheiten 34a und 34b für die Zwei-Weg-Vorausage vergrößert werden.

#### Die elfte Ausführungsform

Fig. 87 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 111 dar, welche eine elfte Ausführungsform der vorliegenden Erfindung zeigt. Bei der in Fig. 87 dargestellten Vorrichtung zur Bildkompression/-expansion ist die mit dem lokalen Datenbus LDB2 verbundene Einheit zur Bewegungsvoraussage 47b der in Fig. 84 gezeigten achten Ausführungsform entfernt.

In diesem Beispiel wird die von der Übertragung des Originalbildes aus der Steuereinheit 27 in den Bildpufferspeicher 51 abweichende Datenübertragung mit den in eine Helligkeitskomponente und eine Farbdifferenzkomponente aufgeteilten Originalbilddaten in der folgenden Art und Weise ausgeführt.

Hinsichtlich der Helligkeitskomponente bezüglich des Bildes für die Innenbildcodierung wird das zu codierende Bild aus dem Bildpufferspeicher 51 zu einer Pixelverarbeitungseinheit 35 übertragen. Das zu reproduzierende Bild wird in den Bildpufferspeicher 51 und einen lokalen Speicher 56a übertragen.

Für das Bild für eine Ein-Weg-Vorausagecodierung werden auf der Seite der Bewegungsvoraussage die Bilddaten im Bereich der Bewegungssuche aus dem lokalen Speicher 56a in die Einheit zur Bewegungsvoraussage 47 übertragen und werden die zu codierenden Bilddaten aus dem Bildpufferspeicher 51 zur Steuereinheit 27 übertragen. Auf der Codierseite werden die vorausgesagten Bilddaten aus dem Pufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen und die zu codierenden Bilddaten aus dem Pufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen. Die zu reproduzierenden Bilddaten werden in den Bildpufferspeicher 51 und den lokalen Speicher 56a übertragen.

Hinsichtlich des Bildes für die Zwei-Weg-Vorausagecodierung werden auf der Seite der Bewegungsvoraussage die Bilddaten im Bereich der Bewegungssuche an zwei Bildern aus dem lokalen Speicher 56a in die Einheit zur Bewegungsvoraussage 47 übertragen, und die zu codierenden Bilddaten werden durch die Steuereinheit 27 aus dem Bildpufferspeicher 51 in die Voraussageeinheit 47 übertragen. Auf der Codierseite werden zwei Bilder der vorausgesagten Bilddaten aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen, und das zu codierende Bild wird aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen.

Hinsichtlich der Farbdifferenzkomponente werden, bezüglich des Bildes für eine Innenbildcodierung, die zu codierenden Bilddaten aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen. Die zu reproduzierenden Bilddaten werden in einen lokalen Speicher 56e übertragen.

Bezüglich eines Bildes für eine Ein-Weg-Vorausagecodierung werden die zu codierenden Bilddaten aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen, und die vorausgesagten Bilddaten werden aus dem lokalen Speicher 56e in die Pixelverarbeitungseinheit 35 übertragen. Die zu reproduzierenden Bilddaten werden in den lokalen Speicher 56e übertragen.

Bezüglich des Bildes für die Zwei-Weg-Vorausagecodierung wird das Farbdifferenzsignal des zu codierenden Bildes aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 35 übertragen, und zwei Bilder der vorausgesagten Bilddaten werden aus dem lokalen Speicher 56e in die Pixelverarbeitungseinheit 35 übertragen. Die Berechnung der Bildübertragungsmenge der entsprechenden Datenbusse bei der Zwei-Weg-Vorausage, bei der die Datenübertragungsmenge maximal wird, erfolgt für ein Bild des Formats 4 : 4 : 4 mit passender Farbdifferenzkomponente folgendermaßen.

Bei der in Fig. 8 dargestellten Vorrichtung zur Bildkompression/-expansion der sechsten Ausführungsform weist die über den Pixeldatenbus PD übertragene Datenübertragungsmenge für das vorstehend genannte Format insgesamt 4806 Pixel auf.

Bei der in Fig. 87 dargestellten Vorrichtung zur Bildkompression/-expansion der elften Ausführungsform werden für das vorstehend genannte Format 4 : 4 : 4 768 Pixel der Originalbilddaten, 578 Pixel der Helligkeitskomponente von zwei Bildern der vorausgesagten Bilddaten und 768 Pixel für die zu codierenden Bilddaten über den Pixeldatenbus PD übertragen, und daher ist die Gesamtdatenübertragungsmenge nicht größer als 2114

Pixel.

Am lokalen Datenbus LDB1 wird die Menge 768 Pixel für eine Ebene der vorausgesagten Bilddaten betragen, wenn der Bereich der Bewegungssuche  $\pm 16$  Pixel umfaßt, und daher beträgt die Datenübertragungsmenge für die zwei Ebenen der vorausgesagten Bilddaten 1536 Pixel.

Über den lokalen Bus LB5 werden nur 578 Pixel der Helligkeitskomponente der vorausgesagten Bilddaten der zwei Ebenen als entsprechende Farbdifferenzkomponenten Cb und Cr übertragen, und daher beträgt die Datenübertragungsmenge insgesamt 1156 Pixel.

Durch separates Codieren der entsprechenden Komponenten der Bilddaten und durch Vorsehen eines zusätzlichen Busses LB5 für die Farbdifferenzkomponenten kann die Datenübertragungsmenge über den Pixeldatenbus PB verkleinert werden. Ferner wird die durch einen Datenbus übertragene Datenmenge verkleinert, da die Datenübertragungsmenge für die entsprechenden Datenbusse gemittelt wird. Daher wird im Vergleich zu der in Fig. 8 dargestellten sechsten Ausführungsform die Datenübertragungsgeschwindigkeit im wesentlichen um den Faktor 2,2 verbessert, und daher kann die Verarbeitungsgeschwindigkeit in der Vorrichtung zur Bildkompression/-expansion vergrößert werden.

#### Die zwölfte Ausführungsform

Fig. 88 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 112 dar, welche eine zwölfte Ausführungsform der vorliegenden Erfindung zeigt. Bei der in Fig. 88 gezeigten zwölften Ausführungsform ist die Pixelverarbeitungseinheit 35 der in Fig. 87 dargestellten elften Ausführungsform in zwei Pixelverarbeitungseinheiten 35a und 35b unterteilt, welchen die Verarbeitung der entsprechenden Helligkeitskomponente und Farbdifferenzkomponente zugewiesen wird.

Folglich wird in der zwölften Ausführungsform die Operation für entsprechende Komponenten durch die Pixelverarbeitungseinheiten 35a und 35b ausgeführt, wobei die Operationsmenge in einer Pixelverarbeitungseinheit verkleinert werden kann, und daher kann die Verarbeitungsgeschwindigkeit in der Vorrichtung zur Bildkompression/-expansion verbessert werden.

#### Die dreizehnte Ausführungsform

Fig. 89 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 113 dar, welche eine dreizehnte Ausführungsform der vorliegenden Erfindung zeigt. Die dreizehnte Ausführungsform umfaßt, zusätzlich zum Aufbau der elften und zwölften Ausführungsform, einen Aufbau, welcher eine Erweiterung des Bereichs der Bewegungssuche ermöglichen kann. Insbesondere werden durch eine Mehrzahl von Pixelverarbeitungseinheiten 36a und 36b separat eine minimal notwendige Helligkeitskomponente aus den aus dem Bildpufferspeicher 51 übertragenen Bilddaten als Referenzdaten zur Bewegungsvoraussage gewonnen. Die gewonnenen Daten werden durch entsprechende lokale Datenbusse LDB1 und LDB2 in lokalen Speichern 56a und 56b gespeichert. Inzwischen gewinnt die Pixelverarbeitungseinheit 35b aus den aus dem Bildpufferspeicher 51 übertragenen Bilddaten die Farbdifferenzkomponente, die zur Bewegungsvoraussage nicht verwendet wird. Die gewonnene Farbdifferenzkomponente wird durch den lokalen Bus LB5 im lokalen Speicher 56e gespeichert. Bei dem vorstehend beschriebenen Aufbau würde die Datenübertragungsmenge über den Pixeldatenbus PB betragen: 768 Pixel der Originalbilddaten, 578 Pixel der Helligkeitskomponente der beiden Ebenen der vorausgesagten Bilddaten und 768 Pixel der zu codierenden Bilddaten, das heißt insgesamt 2114 Pixel.

Wenn der Bereich der Bewegungssuche  $\pm 32$  Pixel beträgt, dann werden die Daten von 1280 Pixeln über jeden der lokalen Datenbusse LDB1 und LDB2 übertragen.

Ferner werden 578 Pixel der Helligkeitskomponente der beiden Ebenen der vorausgesagten Bilddaten als entsprechende Farbdifferenzkomponenten Cb und Cr auf dem lokalen Datenbus LB5 übertragen, und daher werden insgesamt 1156 Pixel übertragen.

Durch den vorstehend beschriebenen Aufbau kann die Datenübertragungsmenge über den Pixeldatenbus PB verkleinert werden, wenn der Bereich der Bewegungssuche vergrößert wird und das Format 4:4:4 als zu verarbeitendes Bildformat gewählt wird, und die Datenübertragungsmenge kann durch die entsprechenden Busse effektiv aufgeteilt werden.

#### Die vierzehnte Ausführungsform

Fig. 90 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 114 dar, welche eine vierzehnte Ausführungsform der vorliegenden Erfindung zeigt. In der vierzehnten Ausführungsform ist ein Zwei-Richtungs-Puffer mit drei Zuständen 80 von 32 Bit ergänzt, um den Pixeldatenbus PB und den lokalen Datenbus LDB zu verbinden. Die Pixelverarbeitungseinheit 37, die nicht mit dem mit dem lokalen Datenbus LDB verbundenen zweiten Datenausgangsport verbunden ist, kann verwendet werden, wie in Fig. 90 dargestellt. Eine Steuereinheit 29 steuert ferner Operationen des Zwei-Richtungs-Puffers mit drei Zuständen 80, einer Bewegungsvoraussage-Einheit 47 und eines lokalen Speichers 56. Referenzbilddaten zur Bewegungsvoraussage werden durch den lokalen Datenbus LDB übertragen. Das Übertragungs-Timing zu dieser Zeit wird ferner durch eine Steuereinheit 21 gesteuert. Der Zwei-Richtungs-Puffer mit drei Zuständen 80 ist dazu in der Lage, gleichzeitig das Lesen von Daten und das Ausgeben von Daten, die gespeichert worden sind, in Synchronisation mit einem aus der Steuereinheit 29 bereitgestellten Uhrsignal auszuführen.

Unter Bezugnahme auf Fig. 90 werden die Bilddatenübertragung beim Codieren eines Bildes und die Bilddatenübertragungsmenge bei der Verarbeitung von einem Makroblock der Bilddaten beschrieben werden. Es wird angenommen, daß die zu codierenden Bilddaten bereits in einem Bildpufferspeicher 51 gespeichert worden sind

und daß die zur Bewegungsermittlung erforderliche Helligkeitskomponente des Referenzbildes (des zu reproduzierenden Bildes) bereits im lokalen Speicher 56 gespeichert worden ist. Die Übertragungsmenge wird unter der Voraussetzung berechnet, daß das Bildformat 4 : 2 : 0 beträgt, der Bereich der Bewegungsermittlung  $\pm 16$  Pixel umfaßt und daß die Struktur eine Bildstruktur aufweist.

5 Zuerst werden die Originalbilddaten durch den Pixeldatenbus PB aus der Steuereinheit 21 in den Bildpufferspeicher 51 übertragen. Zu dieser Zeit beträgt die Menge der übertragenen Bilddaten 384 Pixel.

Anschließend wird zur Bewegungsvoraussage die Helligkeitskomponente Bildes zur Bewegungsvoraussage durch den Pixeldatenbus PB aus dem Bildpufferspeicher 51 in die Steuereinheit 29 und durch einen zugeordneten lokalen Bus LB4 weiter in die Einheit zur Bewegungsvoraussage 47 übertragen. Zu dieser Zeit beträgt die  
10 übertragene Bilddatenmenge 256 Pixel.

Wenn die Bewegungsvoraussage beendet ist und eine Zwischenbildvoraussage auf Grundlage der festgelegten Codierweise unter Verwendung der Resultate auszuführen ist, dann wird das vorausgesagte Bild (das zu reproduzierende Bild) zum Codieren aus dem Bildpufferspeicher 51 durch den Pixeldatenbus PB in die Pixelverarbeitungseinheit 37 übertragen. Zu dieser Zeit beträgt die auf eine Ebene des vorausgesagten Bildes bezogene  
15 übertragene Bilddatenmenge 451 Pixel. Daher müssen zur Zeit der Zwei-Weg-Voraussage 902 Datenpixel übertragen werden.

Danach wird das zu codierende Bild aus dem Bildpufferspeicher 51 durch den Pixeldatenbus PB zur Pixelverarbeitungseinheit 37 übertragen. Zu dieser Zeit beträgt die übertragene Bilddatenmenge 384 Pixel.

Parallel zu den vorstehend genannten vier Bilddaten-Übertragungsprozessen wird zur Bewegungsvoraussage  
20 eine Helligkeitskomponente des Bildes im Bereich der Bewegungssuche aus dem lokalen Speicher 56 durch den lokalen Datenbus LDB zur Einheit zur Bewegungsvoraussage 47 übertragen. Zu dieser Zeit beträgt die auf eine Ebene des Referenzbildes bezogene übertragene Bilddatenmenge 768 Pixel. Daher müssen zur Zeit der Zwei-Weg-Voraussage 1536 Datenpixel für die beiden Ebenen übertragen werden.

Nach Beendigung des Codierens werden die Daten des zu reproduzierenden Bildes bezüglich des Bildes für  
25 die Innenbildcodierung und des Bildes für die Ein-Weg-Voraussage durch den Pixeldatenbus PB aus der Pixelverarbeitungseinheit 37 in den Bildpufferspeicher 51 übertragen. Gleichzeitig werden die Daten des zu reproduzierenden Bildes zum Zwei-Richtungs-Puffer mit drei Zuständen 80 übertragen. Ferner wird das zu reproduzierende Bild durch den lokalen Datenbus LDB aus dem Zwei-Richtungs-Puffer mit drei Zuständen 80 in den lokalen Speicher 56 mit einer Verzögerung um einen Systemzyklus übertragen. Die Steuereinheit 29 steuert die  
30 Datenübertragung durch den Zwei-Richtungs-Puffer mit drei Zuständen 80 in den lokalen Speicher 56 nur dann, wenn die Helligkeitskomponente des zu reproduzierenden Bildes übertragen wird. Zu dieser Zeit beträgt die Menge von übertragenen Bilddaten: 384 Pixel über den Pixeldatenbus PB und 256 Pixel über den lokalen Datenbus LDB.

Wie vorstehend beschrieben, ist es nicht notwendig, daß die Pixelverarbeitungseinheit 37 ein zweites Datenausgangsport aufweist, welches mit dem lokalen Datenbus LDB direkt verbunden ist, und die Busse können in  
35 gleichartiger Weise wie in der in Fig. 83 dargestellten siebenten Ausführungsform aufgeteilt werden, wobei eine Parallelverarbeitung der Datenübertragung ermöglicht wird; da in der Steuereinheit 29 eine Systemsteuereinrichtung (nicht dargestellt) vorgesehen ist, welche die Datenübertragung durch den Zwei-Richtungs-Puffer mit drei Zuständen 80 in den lokalen Speicher 56 nur dann steuert, wenn die Helligkeitskomponente des zu  
40 reproduzierenden Bildes übertragen wird.

Zu dieser Zeit ist die Datenübertragungsmenge ebenso groß wie diejenige der in Fig. 83 dargestellten siebenten Ausführungsform, und sie wird zur Zeit der Zwei-Weg-Voraussage maximal. Insbesondere werden 1926 Datenpixel über den Pixeldatenbus PB und 1536 Datenpixel über den lokalen Datenbus LDB übertragen. Folglich kann durch Aufteilen der Busse die Menge der über einen Bus übertragenen Daten verkleinert werden,  
45 wobei die Datenübertragungsgeschwindigkeit wesentlich verbessert wird und die Verarbeitungsgeschwindigkeit in der Vorrichtung zur Bildkompression/-expansion verbessert wird.

#### Die fünfzehnte Ausführungsform

50 Fig. 91 stellt ein Blockschaltbild einer Vorrichtung zur Bildkompression/-expansion 115 dar, welche eine fünfzehnte Ausführungsform der vorliegenden Erfindung zeigt. Die fünfzehnte Ausführungsform weist einen Aufbau auf, bei welchem Bitstromdaten durch einen Pufferspeicher 6 und eine externe Kommunikationsvorrichtung 94 eingegeben/ausgegeben werden. Die Bitstromdaten, welche durch Codierbilddaten mittels eines in einer Steuereinheit 23 enthaltenen Prozessors für variable Längen (nicht dargestellt) erhalten werden, und Lauf-/Pegeldaten werden im Pufferspeicher 6 zeitweilig gespeichert. Daher ist es möglich, nur die Bitstromdaten nicht  
55 durch einen Host-Computer 93 zu übertragen, so daß die Übertragungsgeschwindigkeit verbessert werden kann.

Insbesondere der Host-Computer 93 ist stark beansprucht, da verschiedene und zahlreiche Prozesse im Host-Computer 93 ablaufen, wie beispielsweise zusätzlich zum Datenübertragungsprozeß eine Anweisungssteuerung. Durch die Kommunikationsvorrichtung 94 außerhalb der Vorrichtung zur Bildkompression/-expansion 215 können Daten aus dem Pufferspeicher 6 zur Kommunikationsvorrichtung 94 übertragen werden, ohne  
60 durch den Host-Computer 93 zu gehen, und daher können die Daten mit hoher Geschwindigkeit übertragen werden, unabhängig von der Beanspruchung des Host-Computers 93.

#### 65 2. Steuereinheit

##### (1) Steuereinheit 21 (Erstes Beispiel)

Fig. 9 stellt ein Blockschaltbild dar, welches ein Beispiel einer Steuereinheit zeigt. Die in Fig. 9 dargestellte Steuereinheit 21 ist in der in den Fig. 1, 5, 6, 7 und 8 gezeigten Steuereinheit 2 verwendbar.

Unter Bezugnahme auf Fig. 9 umfaßt die Steuereinheit 21: einen Prozessor 11 zur Gesamtsteuerung, einschließlich einer Steuerung der Pipelineverarbeitung, einen Prozessor 12 zum Codieren und Decodieren mit variabler Länge, eine Steuereinheit 13 für eine Einheit zur Bewegungsvoraussage (zum Beispiel die in Fig. 1 dargestellte Voraussageeinheit 41), eine Einheit zur Bildformatumwandlung 14 zum Umwandeln des Bildformats, eine Einheit zur Steuerung der Bilddatenübertragung 15, eine Einheit zur Anweisungsübertragung 16 für eine Einheit zur Pixelverarbeitung (zum Beispiel die in Fig. 1 dargestellte Einheit 3), ein Hauptport 17, welches mit einem Host-Bus verbunden ist (dem in Fig. 1 gezeigten Bus HB), ein Bilddatenport 18, das mit einem Pixelbus (dem in Fig. 1 dargestellten Bus PB) verbunden ist, ein VRAM-Port 19, welches mit einem Eingangs-/Ausgangsbus (dem in Fig. 1 dargestellten Bus IOB) verbunden ist, und ein Codedatenport 20, welches mit einem Codedatenbus (dem in Fig. 1 dargestellten Codebus CB) verbunden ist.

In Fig. 9 bezeichnet das Bezugszeichen "DT" im allgemeinen ein Datensignal, "ADR" bezeichnet ein Adreßsignal, und "Sc" bezeichnet ein Steuersignal.

Die entsprechenden Einheiten 12 bis 16 sind mit dem Prozessor zur Gesamtsteuerung 11 durch Busse verbunden und empfangen Anweisungen, die vom Prozessor zur Gesamtsteuerung 11 angewiesen werden. Die Einheiten 12 bis 16 bewirken unabhängig voneinander eine Verarbeitung gemäß den angewiesenen Anweisungen.

Die Steuereinheit 13, die Einheit zur Formatumwandlung 14 und die Einheit zur Steuerung der Bilddatenübertragung 15 sind mit Bussen für Pixeldaten verbunden, und die Übertragung der Pixeldaten wird unter Steuerung der Einheit zur Steuerung der Bilddatenübertragung 15 ausgeführt.

Das Hauptport 17 ist mit dem Prozessor zur Gesamtsteuerung 11 und dem Prozessor für variable Längen 12 verbunden. Wenn zwischen dem Prozessor zur Gesamtsteuerung 11 und dem variablen Prozessor 12 und dem Pufferspeicher 6 und dem in Fig. 1 dargestellten Arbeitsspeicher 7 Daten übertragen werden, dann werden die Daten durch das Hauptport 17 übertragen.

Das Bilddatenport 18 ist mit der Einheit zur Steuerung der Bilddatenübertragung 15 verbunden. Die Datenübertragung zwischen der Einheit zur Steuerung der Bilddatenübertragung 15 und dem in Fig. 1 dargestellten Bildpufferspeicher 51 wird durch das Bilddatenport 18 ausgeführt. Wenn der Bildpufferspeicher 51 der Einheit zur Bewegungsvoraussage 41 oder der Pixelverarbeitungseinheit 3 zugewiesen ist, dann sieht das Bilddatenport 18 ein Steuersignal vor, das anzeigt, ob die Daten gültig sind oder nicht.

Das VRAM-Port 19 ist mit der Einheit zur Bildformatumwandlung 14 verbunden. Die Datenübertragung zwischen der Einheit zur Bildformatumwandlung 14 und dem in Fig. 1 dargestellten Eingabe-/Ausgabespeicher 81 wird durch das VRAM-Port ausgeführt.

Der Prozeß der Bildkompression wird unter Bezugnahme auf die Fig. 1 und 9 beschrieben werden. Es wird angenommen, daß die zu komprimierenden Bilddaten in dem in Fig. 1 gezeigten Bildpufferspeicher 51 gespeichert worden sind. Zuerst überträgt der Prozessor zur Gesamtsteuerung 11 die Bilddaten aus dem Bildpufferspeicher 51 in die Einheit zur Bildformatumwandlung 14 durch Anlegen einer Anweisung zur Bilddatenübertragung an die Einheit zur Steuerung der Bilddatenübertragung 15. Der Prozessor zur Gesamtsteuerung 11 wird durch die Einheit zur Steuerung der Bilddatenübertragung 15 über die Beendigung der Datenübertragung informiert.

Anschließend legt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Bildformatumwandlung an die Einheit zur Bildformatumwandlung 14. Die Einheit zur Bildformatumwandlung 14 führt Operationen zum Umwandeln des Formats des Bildes aus. Insbesondere werden in der Einheit zur Bildformatumwandlung 14 eine Abtastung der Farbdifferenzsignale und die Umwandlung vom RGB-Format in das YUV-Format ausgeführt. Die Beendigung der Datenübertragung wird von der Einheit zur Steuerung der Bilddatenübertragung 15 an den Prozessor zur Gesamtsteuerung 11 signalisiert.

Um Prozesse zur Bewegungsvoraussage und -bestimmung des Codiermodus zu bewirken, überträgt Prozessor zur Gesamtsteuerung 11 Bilddaten zur Bewegungsvoraussage aus dem Bildpufferspeicher 51 zur Voraussagesteuereinheit 13. Diese Datenübertragung wird durch Ausgeben einer Anweisung zur Bilddatenübertragung aus dem Prozessor zur Gesamtsteuerung 11 an die Einheit zur Steuerung der Bilddatenübertragung 15 ausgeführt. Der Prozessor zur Gesamtsteuerung 11 wird von der Einheit zur Steuerung der Bilddatenübertragung 15 über die Beendigung der Datenübertragung informiert.

Der Prozessor zur Gesamtsteuerung 11 gibt, wenn erforderlich, eine Anweisung zur Datenabtastung an die Steuereinheit zur Bewegungsvoraussage 13 aus, wodurch die Steuereinheit zur Bewegungsvoraussage 13 das Abtasten der Bilddaten ausführt, was Gegenstand der Bewegungsvoraussage ist. Der Prozessor zur Gesamtsteuerung 11 wird von der Steuereinheit 13 über die Beendigung des Abtastens der Daten informiert.

Danach gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Modusberechnungsoperation zur Innenbildcodierung an die Steuereinheit zur Bewegungsvoraussage 13 aus. Die Steuereinheit 13 führt eine Operation für berechnete Werte im Innenbild-Codiermodus an denjenigen Daten aus, die Gegenstand der Bewegungsvoraussage sind (Daten, welche abgetastet worden sind, wie gefordert). Der Prozessor zur Gesamtsteuerung 11 wird von der Steuereinheit 13 über die Beendigung dieser Operation informiert.

Ferner gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Datenübertragung an die Einheit zur Steuerung der Bilddatenübertragung 15 aus, und die Einheit zur Steuerung der Bilddatenübertragung 15 überträgt die Bilddaten im Gebiet der Bewegungssuche aus dem Bildpufferspeicher 51 in die Einheit zur Bewegungsvoraussage 41. Der Prozessor zur Gesamtsteuerung 11 wird von der Einheit zur Steuerung der Bilddatenübertragung 15 über die Beendigung der Datenübertragung informiert.

Anschließend legt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Übertragung von Daten zur Bewegungsvoraussage an die Steuereinheit zur Bewegungsvoraussage 13, und die Steuereinheit 13 überträgt die

zu ermittelnden Daten (Daten, die abgetastet werden, falls erforderlich) in die Einheit zur Bewegungsvoraussage 41. Der Prozessor zur Gesamtsteuerung 11 wird von der Einheit zur Steuerung der Bilddatenübertragung 15 über die Beendigung der Datenübertragung informiert.

Nachdem die Bilddaten zur Suche und die Bilddaten zur Bewegungsvoraussage übertragen wurden, startet die Einheit zur Bewegungsvoraussage 41 die Bewegungsvoraussage und erzeugt nach einer Weile ein Ergebnis der Bewegungsvoraussage, das sind die berechneten Daten für den Zwischenbild-Codiermodus.

Nach Ablauf eines bestimmten Zeitabschnitts gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zum Empfangen des Ergebnisses der Bewegungsvoraussage an die Steuereinheit zur Bewegungsvoraussage 13 aus, und die Steuereinheit 13 empfängt aus der Einheit zur Bewegungsvoraussage 41 Daten, welche auf das Ergebnis der Bewegungsvoraussage hinweisen. Der Prozessor zur Gesamtsteuerung 11 wird von der Steuereinheit 13 über die Beendigung des Datenempfangs informiert.

Danach legt der Prozessor zur Gesamtsteuerung 11 den Codiermodus auf Grundlage der berechneten Wertedaten des Innenbild-Codiermodus fest, welche durch die Operation in der Steuereinheit zur Bewegungsvoraussage 13 erhalten werden, und auf Grundlage der berechneten Wertedaten für den Zwischenbild-Codiermodus, welche durch die Operation in der Einheit zur Bewegungsvoraussage 41 erhalten werden. Die Festlegung des Codiermodus wird durch Bezugnahme auf im Pufferspeicher 6 gespeicherte Tabellendaten zur Festlegung des Codiermodus oder durch Vergleichen der berechneten Daten bewirkt.

Nachdem der Codiermodus festgelegt wurde, wird eine Quellencodierung ausgeführt, das heißt ein DCT und eine Quantisierung. Zuerst gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Bilddatenübertragung an die Steuereinheit zur Bilddatenübertragung 15 aus, und die Steuereinheit zur Bilddatenübertragung 15 überträgt die zu codierenden Daten aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 3. Der Prozessor zur Gesamtsteuerung 11 wird von der Steuereinheit zur Bilddatenübertragung 15 über die Beendigung der Datenübertragung informiert.

Danach gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Quellencodierung an die Pixelverarbeitungseinheit 3 durch die Einheit zur Anweisungsübertragung 16 für eine Pixelverarbeitungseinheit aus. Die Pixelverarbeitungseinheit 3 führt die Quellencodierung, das heißt das DCT und die Quantisierung, in Reaktion auf die Anweisung zur Quellencodierung aus. Der Prozessor zur Gesamtsteuerung 11 wird aus der Pixelverarbeitungseinheit 3 durch die Einheit zur Anweisungsübertragung 16 über die Beendigung der Operation informiert.

Nach der Quellencodierung, das heißt nach dem DCT und der Quantisierung, wird eine Codierung mit variabler Länge ausgeführt. Insbesondere gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung aus, welche eine Übertragung der verarbeiteten Daten in die Pixelverarbeitungseinheit 3 durch die Einheit zur Anweisungsübertragung 16 anweist. In Reaktion auf diese Anweisung überträgt die Pixelverarbeitungseinheit 3 die verarbeiteten Daten zum Prozessor für variable Längen 12. Der Prozessor zur Gesamtsteuerung 11 wird von der Pixelverarbeitungseinheit 3 durch die Einheit zur Anweisungsübertragung 16 über die Beendigung der Datenübertragung informiert.

Um das Codieren mit variabler Länge, das heißt eine Entropiecodierung, auszuführen, gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung zur Entropiecodierung an den Prozessor für variable Längen 12 aus. Der Prozessor für variable Längen 12 führt die Entropiecodierung aus. Bei der Entropiecodierung wird eine den Huffman-Code verwendende Codeumwandlung und dergleichen bewirkt. Insbesondere der Prozessor für variable Längen 12 führt die Codeumwandlung der durch Quellencodierung erhaltenen Daten unter Bezugnahme auf Daten einer im Arbeitsspeicher 7 gespeicherten Codeumwandlungstabelle aus. Der Prozessor zur Gesamtsteuerung 11 wird vom Prozessor für variable Längen 12 über die Beendigung der Entropiecodierung informiert.

Nach Beendigung der Entropiecodierung gibt der Prozessor zur Gesamtsteuerung 11 eine Anweisung aus, welche eine Übertragung der verarbeiteten Daten in den Prozessor für variable Längen 12 anweist. Der Prozessor für variable Längen 12 überträgt die verarbeiteten Daten in den Pufferspeicher 6. Der Prozessor zur Gesamtsteuerung 11 wird vom Prozessor für variable Längen 12 über die Beendigung der Datenübertragung informiert.

Wie aus der vorstehenden Beschreibung zu ersehen ist, kann die Bilddatenkompression sowohl in verschiedenen Verarbeitungseinheiten 13, 14, 15 und 16 in der Steuereinheit 21 als auch im Prozessor für variable Längen 12 unter der Gesamtsteuerung des Steuerprozessors 11 unabhängig ausgeführt werden. Insbesondere können die Umwandlung des Bildformats durch die Einheit zur Bildformatumwandlung, die Bewegungsvoraussage und die Operation für den Codiermodus-Berechnungswert durch die Einheit zur Bewegungsvoraussage 41 und die Steuereinheit zur Bewegungsvoraussage 13, die Bilddatenübertragung durch die Steuereinheit zur Bilddatenübertragung 15, die Quellencodierung durch die Pixelverarbeitungseinheit 3 und die Entropiecodierung durch den Prozessor für variable Längen 12 parallel und unabhängig voneinander ausgeführt werden. Daher beeinflusst der Prozessor zur Gesamtsteuerung 11 bei der Steuerung der Pipelineverarbeitung die vorstehend beschriebenen Prozessoren für alle Makroblockdaten des Bildes. Die Pipelineverarbeitung der vorstehend beschriebenen Prozesse wird unter Bezugnahme auf die Fig. 11 und 12 nachstehend detailliert beschrieben. Die Pipelineverarbeitung realisiert eine hocheffektive Bildkompression.

Eine Bildexpansion wird in gleichartiger Weise wie die Bildkompression ebenfalls ausgeführt, und verschiedene Prozesse in der Steuereinheit 21 werden unabhängig voneinander unter der Gesamtsteuerung durch den Prozessor zur Gesamtsteuerung 11 ausgeführt. Bei der Bildexpansion werden die im Pufferspeicher 6 gespeicherten Codedaten invers codiert, und schließlich werden die expandierten Bilddaten durch die Einheit zur Bildformatumwandlung 14 in den in Fig. 1 dargestellten Eingabe-/Ausgabespeicher 81 übertragen.

## (2) Steuereinheit 22 (Zweites Beispiel)

Fig. 10 stellt ein Blockschaltbild dar, welches ein anderes Beispiel der Steuereinheit zeigt. Die in Fig. 10 dargestellte Steuereinheit 22 ist in der in Fig. 4 gezeigten Steuereinheit 22 verwendbar. Im Vergleich zu der in Fig. 9 dargestellten Steuereinheit 21 ist die Steuereinheit zur Bewegungsvoraussage 13 aus der Steuereinheit 22 entfernt. Insbesondere die in Fig. 4 dargestellte Vorrichtung zur Bildkompression/-expansion 102 erfordert keine Bewegungsvoraussage, und die Steuereinheit zur Bewegungsvoraussage aus der Steuereinheit 22 ist entsprechend entfernt.

Es wird bemerkt, daß außer der Bewegungsvoraussage die in Fig. 10 dargestellte Steuereinheit 22 in gleichartiger Weise wie die in Fig. 9 dargestellte Steuereinheit 21 betrieben wird. Folglich können gleichartige Vorteile wie bei der in Fig. 9 dargestellten Steuereinheit 21 erhalten werden, das heißt, die verschiedenen internen Einheiten in der Steuereinheit 22 können unabhängig voneinander und parallel zueinander betrieben werden, und die Bilddatenkompression kann unter der Steuerung der Pipelineverarbeitung wirkungsvoll ausgeführt werden.

## (3) Pipelineverarbeitung

Der in den Fig. 9 und 10 dargestellte Prozessor zur Gesamtsteuerung 11 führt die Pipelineverarbeitung in der Vorrichtung zur Bildkompression/-expansion in der folgenden Art und Weise aus.

Fig. 11 stellt ein Zeitablaufdiagramm dar, das die Pipelineverarbeitung zeigt, welche unter der Steuerung des in Fig. 9 dargestellten Prozessors zur Gesamtsteuerung 11 ausgeführt wird. Unter Bezugnahme auf Fig. 11 stellt die Abszisse die Zeit und die Ordinate die Prozessschritte zur Bildkompression dar. Bei der Bildkompression werden zu verarbeitende Ein-Bild-Daten in eine Anzahl von Makroblöcke MB eingeteilt und die Prozesse zur Bildkompression an allen Makroblockdaten ausgeführt. Im in Fig. 11 dargestellten Beispiel werden die Daten von zehn Makroblöcken MB1 bis MB10 verarbeitet.

Unter Bezugnahme auf Fig. 11 wird die Verarbeitung der Daten von einem Makroblock MB1 beispielhaft beschrieben werden. Zuerst werden in einem Zeitabschnitt T1 die RGB-Daten der Makroblockdaten MB1 aus dem in Fig. 1 dargestellten Eingabe-/Ausgabespeicher 81 zur Steuereinheit 2 übertragen (Schritt S1). In einem Zeitabschnitt T2 werden die RGB-Daten in YUV-Daten durch die in Fig. 9 dargestellte Einheit zur Bildformatumwandlung 14 umgewandelt (Schritt S2).

In einem Zeitabschnitt T3 werden die YUV-Daten in den Bildpufferspeicher 51 übertragen, und die Bilddaten, welche Gegenstand der Suche sind, werden zur Steuereinheit 2 übertragen. Zur gleichen Zeit werden die Bilddaten, welche Gegenstand der Suche sind, in die Einheit zur Bewegungsvoraussage 41 durch den Pixeldatenbus PB übertragen (Schritt S3).

In einem Zeitabschnitt T4 wird durch die Einheit zur Bewegungsvoraussage 41 die Bewegungsvoraussage bewirkt (Schritt S4). In einem Zeitabschnitt T5 wird durch die Steuereinheit 2 ein Codiermodus festgelegt (Schritt S5).

In einem Zeitabschnitt T6 werden die zu codierenden Bilddaten aus dem Bildpufferspeicher 51 in die Pixelverarbeitungseinheit 3 übertragen, während die Referenzbilddaten zur Pixelverarbeitungseinheit 3 durch den Pixeldatenbus PB übertragen werden (Schritt S6).

In einem Zeitabschnitt T7 wird die Bildverarbeitung für das Codieren, das heißt das DCT und die Quantisierung, durch die Pixelverarbeitungseinheit 3 bewirkt (Schritt S7). In einem Zeitabschnitt T8 werden die verarbeiteten Daten, das heißt die Lauf-/Pegeldaten, zum Pufferspeicher 6 durch den Codebus CB übertragen (Schritt S8).

In einem Zeitabschnitt T9 wird das Codieren mit variabler Länge durch den in Fig. 9 dargestellten Prozessor für variable Längen 12 bewirkt (Schritt S9). In einem Zeitabschnitt T10 werden die verarbeiteten Daten, das heißt die Codedaten variabler Länge, durch den Codebus CB in den Pufferspeicher 6 übertragen (Schritt S10).

Wie vorstehend beschrieben, wird in der Vorrichtung zur Bildkompression/-expansion 101 die Pipelineverarbeitung unter der Steuerung durch den Prozessor zur Gesamtsteuerung 11 ausgeführt. Wie in Fig. 11 dargestellt, werden daher die entsprechenden Prozesse in den Zeitabschnitten T2 bis T10 nacheinander an anderen Makroblockdaten MB2 bis MB10 in gleichartiger Weise wie an den Makroblockdaten MB1 ausgeführt.

Fig. 12 zeigt einen Parallelbetrieb in dem in Fig. 11 dargestellten einen Zeitabschnitt T10. Insbesondere zeigt Fig. 11 im Zeitabschnitt T10 den Betriebszustand verschiedener interner Einheiten in der Steuereinheit 21 und der Vorrichtung zur Bildkompression/-expansion 101, welche in den Fig. 1 und 9 dargestellt sind. Unter Bezugnahme auf Fig. 12 überträgt im Zeitabschnitt T10 der Eingabe-/Ausgabebus IOB die RGB-Daten des zehnten Makroblocks MB10 (der in Fig. 11 dargestellte Schritt S1). Der Pixeldatenbus PB überträgt die Daten des achten und des fünften Makroblocks MB8 und MB5 (Schritte S3 und S6). Der Codebus CB überträgt die Daten des dritten und des ersten Makroblocks MB3 und MB1 (Schritte S8 und S10).

Der Prozessor zur Gesamtsteuerung 11 steuert die Übertragung der RGB-Daten des zehnten Makroblocks MB10 (Schritt S3) und legt parallel dazu den Codiermodus des sechsten Makroblocks MB6 fest (Schritt S5). Ferner bewirkt der Prozessor zur Gesamtsteuerung 11 die Reihenfolgesteuerung der Makroblöcke MB1 bis MB10, das heißt die Steuerung für die Pipelineverarbeitung.

Der Prozessor für variable Längen 12 bewirkt das Codieren mit variabler Länge an dem zweiten Makroblock MB2 (Schritt S9). Die Einheit zur Anweisungsübertragung 16 führt die Bildverarbeitungs-Steuerung des vierten Makroblocks MB4 aus (Schritt S7).

Die Steuereinheit zur Bewegungsvoraussage 13 führt die Bewegungsvoraussage des siebenten Makroblocks MB7 aus (Schritt S4). Die Einheit zur Steuerung der Bilddatenübertragung 15 steuert die Datenübertragung des achten und fünften Makroblocks MB8 und MB5 (Schritt S8).



Die Einheit zur Bilddatenumwandlung 14 führt die Formatumwandlung (von RGB zu YUV) am neunten Makroblock MB9 aus (Schritt S2). Die Pixelverarbeitungseinheit 3 führt die Bildverarbeitung am vierten Makroblock MB4 aus (Schritt S7). Die Einheit zur Bewegungsvoraussage 4 führt die Voraussage des siebenten Makroblocks MB7 aus (Schritt S4).

5

#### (4) Aufbau der entsprechenden internen Einheiten

##### (i) Prozessor zur Gesamtsteuerung 11 (Fig. 13)

Fig. 13 stellt ein Blockschaltbild des in Fig. 9 gezeigten Prozessors zur Gesamtsteuerung 11 dar. Unter Bezugnahme auf Fig. 13 umfaßt der Prozessor zur Gesamtsteuerung 11: einen Anweisungsspeicher 111, eine Programmsteuereinrichtung 112, eine Reihenfolgesteuereinrichtung/einen Adreßoperator 113, eine Systemsteuereinrichtung 114, ein Operationsregisterfile 115 und einen Datenoperator 116. In dieser Figur bezeichnet das Bezugszeichen "DT" Daten, "ADR" bezeichnet ein Adreßsignal, "ID" bezeichnet Anweisungsdaten, "SSc" bezeichnet ein Status-Steuersignal und "Sc" bezeichnet ein Steuersignal.

15

##### (ii) Einheit zur Steuerung der Bewegungsvoraussage 13 (Fig. 14)

Fig. 14 stellt ein Blockschaltbild einer in Fig. 9 dargestellten Einheit zur Steuerung der Bewegungsvoraussage 13 dar. Unter Bezugnahme auf Fig. 14 umfaßt die Einheit zur Steuerung der Bewegungsvoraussage 13: eine Modus-/Timing-Steuereinrichtung 131 für die Einheit zur Bewegungsvoraussage, eine Gesamtsteuereinrichtung 132, einen Vektordatenprozessor 133, einen Prozessor 134 für Bilddaten für die Bewegungsermittlung und einen Berechnungsoperator 135 für den Innenbild-Codiermodus.

Die Modus-/Timing-Steuereinrichtung 131 sieht ein Steuersignal Smc für die Einheit zur Bewegungsvoraussage 41 vor. Der Vektordatenprozessor 133 sieht ferner ein Steuersignal Smc für die Einheit zur Bewegungsvoraussage 41 vor. Der Vektordatenprozessor 133 empfängt die die Ergebnisse der Bewegungsermittlung anzeigenden Daten Dmr aus der Einheit zur Bewegungsvoraussage 41. Der Prozessor 134 für Bilddaten für die Bewegungsermittlung sieht ein Steuersignal Smc für Bilddaten Dmi, welche Gegenstand der Bewegungsermittlung sind, und für die Einheit zur Bewegungsvoraussage 41 vor. Der Prozessor 134 für Bilddaten für die Bewegungsermittlung gibt Pixeldaten PD an den Berechnungsoperator 135 für den Innenbild-Codiermodus aus.

30

##### (iii) Einheit zur Bildformatumwandlung 14 (Fig. 15)

Fig. 15 stellt ein Blockschaltbild der in Fig. 9 gezeigten Einheit zur Bildformatumwandlung 14 dar. Unter Bezugnahme auf Fig. 15 umfaßt die Einheit zur Bildformatumwandlung 14: eine Gesamtsteuereinrichtung 141, eine VRAM-Steuereinrichtung 142, einen Farbkonverter 143, ein UV-Filter 144 und einen Bildspeicher 145.

Die Gesamtsteuereinrichtung 141 empfängt Daten DT und ein Adreßsignal ADR durch einen Hauptdatenbus MB. Die VRAM-Steuereinrichtung 142 ist mit einem VRAM-Port 19 verbunden. Der Bildspeicher 145 empfängt Pixeldaten PD durch einen internen Pixeldatenbus IPB und stellt gespeicherte Daten bereit.

40

##### (iv) Einheit zur Steuerung der Bilddatenübertragung 15 (Fig. 16)

Fig. 16 stellt ein Blockschaltbild der in Fig. 9 dargestellten Einheit zur Steuerung der Bilddatenübertragung 15 dar. Unter Bezugnahme auf Fig. 16 umfaßt die Einheit zur Steuerung der Bilddatenübertragung 15: eine Gesamtsteuereinrichtung 151, eine Steuereinrichtung für den Bilddatenspeicher 152, eine Steuereinrichtung für die Bilddatenübertragung 153 und eine Steuereinrichtung für die Pixeldatenbusübertragung 154.

In Fig. 16 bezeichnet das Bezugszeichen "Simc" ein Bilddatenspeicher-Steuersignal, "Sptc" bezeichnet ein Bilddatenübertragungs-Steuersignal für die Pixelverarbeitungseinheit 3, und "Smtc" bezeichnet ein Bilddatenübertragungs-Steuersignal für die Einheit zur Bewegungsvoraussage 41.

50

##### (v) Einheit zur Anweisungsübertragung 16 (Fig. 17)

Fig. 17 stellt ein Blockschaltbild der in Fig. 9 gezeigten Einheit zur Anweisungsübertragung 16 dar. Unter Bezugnahme auf Fig. 17 umfaßt die Einheit zur Anweisungsübertragung 16 für die Pixelverarbeitungseinheit 3: eine Gesamtsteuereinrichtung 161, eine Steuereinrichtung zur Anweisungsübertragung 162 und einen Statusmonitor 163. Die Steuereinrichtung zur Anweisungsübertragung 162 sieht ein Steuersignal Spc für die Pixelverarbeitungseinheit 3 vor. Der Statusmonitor 163 empfängt ein Statussignal Sps aus der Pixelverarbeitungseinheit 3. Die Gesamtsteuereinrichtung 161 empfängt Daten DT und ein Adreßsignal ADR durch einen Hauptdatenbus MB.

60

Auf diese Weise empfangen die in den Fig. 14 bis 17 dargestellten entsprechenden internen Einheiten 13, 14, 15 und 16 aus dem in Fig. 13 dargestellten Prozessor zur Gesamtsteuerung 11 Signale, welche den Start von entsprechenden Prozessen anweisen, und übertragen Steuersignale, welche auf die Beendigung der entsprechenden Prozesse hinweisen, zurück zum Prozessor zur Gesamtsteuerung 11, wenn die entsprechenden Prozesse abgeschlossen sind. Es wird ausdrücklich darauf hingewiesen, daß die internen Einheiten 13, 14, 15 und 16 die entsprechenden Prozesse unabhängig voneinander in Reaktion auf die Prozeßstartsignale aus dem Prozessor zur Gesamtsteuerung 11 ausführen können. Folglich kann unter der Steuerung des Prozessors zur Gesamtsteuerung 11 die Datenverarbeitung zur Bildkompression und Bildexpansion wirksam pipelineartig ausgeführt werden.

65



## (5) Steuereinheit 200 (Drittes Beispiel)

Fig. 92 stellt ein Blockschaltbild dar, welches ein weiteres Beispiel der Steuereinheit zeigt. Die in Fig. 92 dargestellte Steuereinheit 200 kann in der in den Fig. 1, 5 bis 8 und 84 bis 91 dargestellten Steuereinheit verwendet werden.

Unter Bezugnahme auf Fig. 92 umfaßt die Steuereinheit 200: einen Host-Schnittstellenabschnitt 201, der mit einem Host-Bus HB verbunden ist; einen Abschnitt zur Gesamtsteuerung 202 zum Ausführen verschiedener Steuerungen, einschließlich der Pipelinesteuerung; einen Video- Eingangs- /Ausgangsabschnitt 203' der mit einem Eingangs-/Ausgangsbuss IOB verbunden ist; einen Hauptprozessor 204, der in Macrocodes verfaßte Programme ausführt; einen Bildspeicher-Steuerabschnitt 205; einen Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 zum Steuern eines Bildspeichers zur Bewegungsvoraussage; einen Codedaten-Steuerabschnitt 207, welcher die Codedaten in eine und aus einer Pixelverarbeitungseinheit überträgt; einen Parameter-Operationsabschnitt 208 und einen Prozessor für variable Längen 209 zum Codieren/Decodieren mit variabler Länge.

Ferner umfaßt die Steuereinheit 200 als interne Busse: einen ersten internen Bus 210, der zwischen dem Video-Eingangs-/Ausgangsabschnitt 203 und dem Bildspeicher-Steuerabschnitt 205 geschaltet ist; einen zweiten internen Bus 211, der jeweils mit dem Bildspeicher-Steuerabschnitt 205, dem Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 und mit dem Parameter-Operationsabschnitt 208 verbunden ist; einen dritten internen Bus 212, der zwischen dem Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 und dem Bildspeicher-Steuerabschnitt 205 geschaltet ist; einen vierten internen Bus, der zwischen dem Codedaten-Steuerabschnitt 207 und dem Prozessor für variable Längen 209 geschaltet ist; und einen Hauptbus 214, der mit allen Blöcken verbunden ist.

Der Aufbau und die Betriebsweise der verschiedenen Blöcke wird im folgenden beschrieben werden.

Der Host-Schnittstellenabschnitt 201 stellt einen Steuerabschnitt dar, welcher die Datenübertragung zwischen dem externen Host-Computer und der Steuereinheit 200 steuert. Der Dateneingang/-ausgang durch den Host-Schnittstellenabschnitt 201 wird durch den Hauptbus 214 übertragen, welcher mit jedem Block in der Steuereinheit 200 verbunden ist.

Der Abschnitt zur Gesamtsteuerung 202 stellt einen Block dar, welcher vor allem verschiedene Operations-Timings und Pipelineverarbeitungen in der Steuereinheit 200 überwacht. Zur Zeit der Bildcodierung erzeugt der Abschnitt zur Gesamtsteuerung 202 einen internen Synchronisierungspuls auf Grundlage der Bildeinheit oder der Makroblockeinheit, welcher auf einer Basisuhr und einem Synchronisierungssignal basiert, steuert die Hauptparameter beim Codieren, überträgt Daten und so weiter.

Der Video-Eingangs-/Ausgangsabschnitt 203 weist beispielsweise eine Schnittstelle mit dem in Fig. 83 dargestellten Eingangs-/Ausgangsspeicher 81 auf, und in der Steuereinheit 200 ist er mit dem Bildspeicher-Steuerabschnitt 205 durch den ersten internen Bus 210 verbunden.

Der Hauptprozessor 204 stellt einen Block dar, der vor allem für die angepaßte Verarbeitung beim Codieren verantwortlich ist, dessen Details nachstehend beschrieben werden. Wenn ein Konflikt mit dem Host-Schnittstellenabschnitt 201 vorhanden ist, dann wird der Hauptprozessor 204 der Master des Hauptbusses 214 sein und die Datenübertragung des Registers/Speichers in jedem Block durchführen.

Der Bildspeicher-Steuerabschnitt 205 umfaßt als externe Schnittstelle eine Schnittstelle von 32 Bit, welche als Master eines 32 Bit-Pixeldatenbusses (beispielsweise des Pixeldatenbusses PB der Fig. 83) zwischen einem Bildpufferspeicher (zum Beispiel dem in Fig. 83 dargestellten Bildpufferspeicher 51) und einer Pixelverarbeitungseinheit (beispielsweise der in Fig. 83 dargestellten Pixelverarbeitungseinheit 33) wirkt. Der Bildspeicher-Steuerabschnitt 205 ist in der Steuereinheit 200 mit dem Video-Eingangs-/Ausgangsabschnitt 203 durch den ersten internen Bus 210, mit dem Parameter-Steuerabschnitt 208 sowie dem Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 durch den zweiten internen Bus 211 und ferner mit einer anderen Schnittstelle des Steuerabschnitts für die Einheit zur Bewegungsvoraussage 206 durch den dritten internen Bus 212 verbunden.

Der Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 umfaßt als externe Schnittstellen eine Schnittstelle, welche ein Steuersignal an die Einheit zur Bewegungsvoraussage (zum Beispiel die in Fig. 83 dargestellte Einheit zur Bewegungsvoraussage 47) ausgibt, und eine Schnittstelle, welche ein eine Adresse enthaltendes Steuersignal an einen lokalen Speicher (zum Beispiel den in Fig. 83 dargestellten lokalen Speicher 56) ausgibt.

Der Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 umfaßt ferner eine Schnittstelle mit einem lokalen Bus (beispielsweise dem in Fig. 83 dargestellten lokalen Bus LB4) zum Übertragen von Parametern, wie beispielsweise den Bewegungsvektor- und Musterdaten (Bilddaten), zu und aus der Einheit zur Bewegungsvoraussage. Ferner ist der Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 in der Steuereinheit 200 mit dem Bildspeicher-Steuerabschnitt 205 sowie dem Parameter-Steuerabschnitt 208 durch den zweiten internen Bus 211 und mit dem Bildspeicher-Steuerabschnitt 205 durch den dritten internen Bus 212 verbunden.

Der Codedaten-Steuerabschnitt 207 weist eine Schnittstelle auf, die mit einem Codebus (beispielsweise dem in Fig. 83 gezeigten Codebus CB) verbunden ist zur Verbindung mit einer Pixelverarbeitungseinheit (beispielsweise der in Fig. 83 dargestellten Pixelverarbeitungseinheit 33) und mit einem Pufferspeicher (zum Beispiel dem in Fig. 83 dargestellten Pufferspeicher 6). Der Codedaten-Steuerabschnitt 207 ist in der Steuereinheit 200 mit dem Prozessor für variable Längen 209 durch den vierten internen Bus 213 verbunden.

Der Parameter-Operationsabschnitt 208 ist mit dem Bildspeicher-Steuerabschnitt 205 und mit dem Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 durch den zweiten internen Bus 211 verbunden. Durch den zweiten internen Bus 211 werden die Musterdaten aus dem Bildspeicher-Steuerabschnitt 205 in den Parameter-Operationsabschnitt 208 ausgegeben, und der Parameter-Operationsabschnitt 208 gewinnt Parameter bezüglich des Musters. Der Parameter-Operationsabschnitt 208 umfaßt ein Register, welches das Ergebnis der Operation

darin speichert. Das im Register gespeicherte Operationsergebnis kann durch den Hauptbus 214 übertragen (gelesen) werden (Slave-Operation).

Der Prozessor für variable Längen 209 ist mit dem Codedaten-Steuerabschnitt 207 durch den vierten internen Bus 213 verbunden. Das Codieren mit variabler Länge von Daten des Lauf-/Pegeltyps oder des Quantisierungsindex, welche aus dem Codedaten-Steuerabschnitt 207 übertragen werden, wird im Prozessor für variable Längen 209 bewirkt. Die im Prozessor für variable Längen 209 erzeugten Bitstromdaten werden in den Bildpufferspeicher (beispielsweise den in Fig. 83 dargestellten Bildpufferspeicher 51) übertragen.

Das Wesen der Prozesse in den entsprechenden Blöcken in der Steuereinheit 200 während der Bildcodierung wird unter Bezugnahme auf die Fig. 83 und 92 detailliert beschrieben werden.

Die Steuereinheit 23 hat die Aufgabe, die Pixelverarbeitungseinheit 33 und die Einheit zur Bewegungsvoraussage 47 zu steuern, um das Codieren auszuführen. Daher werden vor allem die folgenden vier Datensteuertypen ausgeführt.

(1) Lesen der Originaldaten und Speichern der Daten im Bildpufferspeicher 51.

(2) Lesen der Musterdaten zur Bewegungsvoraussage und Aktivierung der Einheit zur Bewegungsvoraussage 47.

(3) Lesen und Festlegen der Ergebnisse der Bewegungsvoraussage und Aktivieren der Pixelverarbeitungseinheit 33.

(4) Lesen und Verarbeiten einer variablen Länge der Codedaten (Quantisierungsindex/Laufpegel) und Schreiben in den Pufferspeicher 6.

Die vier Datensteuerungstypen werden unter Bezugnahme auf die Operationen der entsprechenden Blöcke beschrieben werden.

#### (i) Lesen der Originaldaten und Schreiben der Daten in den Bildpufferspeicher 51 (Abbilden)

Die in einem externen Eingangs-/Ausgangsspeicher gespeicherten Originaldaten (die zu codierenden Originalbilddaten) werden auf Grundlage einer sogenannten Makroblockeinheit segmentiert und zum Video-Eingangs-/Ausgangsabschnitt 203 übertragen.

Der Video-Eingangs-/Ausgangsabschnitt 203 umfaßt darin eine Schaltung zur Umwandlung aus dem RGB-Format in das YUV-Format sowie eine Filterschaltung, und die Formatumwandlung wird an dieser Stelle ausgeführt (zum Beispiel: RGB  $\rightarrow$  YUV, 4 : 2 : 2  $\rightarrow$  4 : 2 : 0 usw.). Die verarbeiteten Daten werden durch den ersten internen Bus 210 zum Bildspeicher-Steuerabschnitt 205 übertragen. Der Bildspeicher-Steuerabschnitt 205 bildet die übertragenen Daten in einem Format ab (schreibt sie in einem Format), welches für das folgende Codieren im durch den Bildspeicher-Steuerabschnitt 205 gesteuerten Bildpufferspeicher 51 geeignet ist. Das Verfahren zum Abbilden wird nachstehend detailliert beschrieben werden.

Die Erzeugung einer Adresse des Bildpufferspeichers 51 wird durch den Bildspeicher-Steuerabschnitt 205 ausgeführt. Die Steuerung des Bildpufferspeichers 51 und das Timing der Datenübertragung aus dem Video-Eingangs-/Ausgangsabschnitt 203 wird durch den Abschnitt zur Gesamtsteuerung 202 überwacht.

#### (ii) Lesen der Musterdaten zur Bewegungsvoraussage und Steuern der Einheit zur Bewegungsvoraussage

Um die Bewegungsvoraussage eines zu codierenden Makroblocks zu bewirken, liest der Bildspeicher-Steuerabschnitt 205 die dem Makroblock entsprechenden Bilddaten aus dem Bildpufferspeicher 51. Die Daten werden als Musterdaten zugewiesen. Der Bildspeicher-Steuerabschnitt 205 überträgt die Musterdaten zum Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 und zum Parameter-Operationsabschnitt 208.

Der Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 überträgt die Musterdaten in die Einheit zur Bewegungsvoraussage 47 durch den lokalen Bus LB4 und überträgt gleichzeitig den Musterdaten entsprechende Suchfensterdaten zum lokalen Speicher 56 durch den lokalen Datenbus LDB. Ferner aktiviert der Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 eine Operation zur Bewegungsvoraussage der Musterdaten durch Ausgeben als Eingang, als Steuersignal, wie beispielsweise ein Signal zur Aktivierungssteuerung, an die Einheit zur Bewegungsvoraussage 47.

Ferner überträgt der Bildspeicher-Steuerabschnitt 205 die Musterdaten zum Steuerabschnitt für die Einheit zur Bewegungsvoraussage 206 durch den zweiten internen Bus 111 und gleichzeitig zum Parameter-Operationsabschnitt 208. Der Parameter-Operationsabschnitt 208 gewinnt die Merkmale der Musterdaten durch eine vorgeschriebene Operation, welche Merkmale zur Festlegung des Codierens und dergleichen später verwendet werden. Die an dieser Stelle berechneten Parameter umfassen: eine Varianz (nur des Helligkeitssignals) von jedem Block (im Feld/Bild beide Blöcke) des Musters zur Aktivierungsberechnung, einen Mittelwert und einen Varianzwert des Helligkeitssignals des Muster-Makroblocks und so weiter, welche Pixeloperationen erfordern.

#### (iii) Lesen des Ergebnisses der Bewegungsvoraussage und Aktivieren der Pixelverarbeitungseinheit

Nach mehreren Makroblockzyklen wird aus der Aktivierung der Einheit zur Bewegungsvoraussage das Ergebnis der Bewegungsvoraussage, das heißt, werden die verschiedenen Modi zur Bewegungsvoraussage, Rückbewegungsvektoren in der horizontalen und vertikalen Richtung und berechnete Funktionen derselben, in den Abschnitt zur Steuerung der Einheit zur Bewegungsvoraussage 206 durch den lokalen Bus LB4 aufgenommen.

Der Abschnitt zur Steuerung der Einheit zur Bewegungsvoraussage 206 bestimmt den Voraussagemodus und

den Bewegungsvektor unter Verwendung der vorstehend genannten Berechnungsfunktion. Der bestimmte Bewegungsvektor wird zum Bildspeicher-Steuerabschnitt 205 durch den internen Bus 212 übertragen. Der Bildspeicher-Steuerabschnitt 205 berechnet eine Adresse des vorausgesagten Bildes auf Grundlage des Eingangs-Bewegungsvektors. Der Bildspeicher-Steuerabschnitt 205 steuert den Bildpufferspeicher 51 und die Pixelverarbeitungseinheit 33 auf Grundlage der berechneten Adresse und überträgt die dem Bewegungsvektor entsprechenden vorausgesagten Bilddaten zur Pixelverarbeitungseinheit 33. 5

Ferner übermittelt der Bildspeicher-Steuerabschnitt 205 die Musterdaten zur Pixelverarbeitungseinheit 33 in der gleichartigen vorstehend beschriebenen Weise (Da sich die Pipeline von derjenigen der vorstehend genannten Musterdaten unterscheidet, ist es erforderlich, neu zu übertragen.) und veranlaßt durch Aktivieren der Pixelverarbeitungseinheit 33 die Pixelverarbeitungseinheit 33 dazu, das DCT und die nachfolgenden Prozesse auszuführen. An dieser Stelle werden der Befehl zur Aktivierung und die Parameter, die zum Codieren in der Pixelverarbeitungseinheit 33 (zum Beispiel die Hälfte pro Filteranweisung) erforderlich sind, in ein vorgeschriebenes Register in der Pixelverarbeitungseinheit 33 durch den Codedaten-Steuerabschnitt 207 oder in einen Host-Schnittstellenabschnitt 201 durch die gleiche Pipeline gesetzt. Die Pixelverarbeitungseinheit 33 ist dazu ausgelegt, daß sie von beiden der vorstehend genannten Ports gesetzt werden kann. 10 15

#### (iv) Lesen und Verarbeiten einer variablen Länge der Codedaten und Schreiben in den Pufferspeicher

Die in der Pixelverarbeitungseinheit 33 verarbeiteten Daten (das heißt Daten, die einem Filtern, dem DCT, der Quantisierung oder dergleichen unterworfen worden sind) werden durch die Steuerung des Abschnitts zur Gesamtsteuerung 202 zeitweilig in den Pufferspeicher 6 durch den Codebus CB geschrieben oder direkt zum Codedaten-Steuerabschnitt 207 der Steuereinheit 23 übertragen. Das Datenformat stellt zu dieser Zeit entweder einen Quantisierungsindex (Daten nach der Quantisierung) oder Laufpegeldaten (den in einen Laufpegel umgewandelten Quantisierungsindex) dar. 20

Der Codedaten-Steuerabschnitt 207 empfängt die Daten bei einem beliebigen Timing und überträgt dieselben durch den vierten internen Bus 213 zum Prozessor für variable Längen 209. Der Prozessor für variable Längen 209 führt eine sogenannte Huffman-Codierung gemäß einem Unterprogramm aus und schreibt die Ergebnisse in den externen Pufferspeicher 6. Zur Zeit der Codeumwandlung, wie beispielsweise der Huffman-Codierung, ist es möglich, auf im Arbeitsspeicher 7 gespeicherte Tabellendaten zuzugreifen. 25

Im folgenden wird beschrieben werden, wie die Hauptverarbeitung 204 beim Codieren in der vorstehend beschriebenen Verarbeitung beteiligt ist. 30

Wenn eine Bestimmung auszuführen ist, die sich von der im Abschnitt zur Steuerung der Einheit zur Bewegungsvoraussage 206 ausgeführten Bestimmung unterscheidet, wenn beispielsweise eine Benutzerin wünscht, ihr eigenes Bestimmungsverfahren zu verwenden, dann schreitet der Prozeß in folgender Weise fort. Der Hauptprozessor 204 greift durch den Hauptbus 214 auf ein Register zu, welches Vektoren und berechnete Werte für verschiedene Voraussagemodi im Abschnitt zur Steuerung der Einheit zur Bewegungsvoraussage 206 speichert. Die Vektoren und die berechneten Werte werden einmalig in Makrocodes verarbeitet und an das Originalregister in einem vorbestimmten Zeitabschnitt zurückgegeben. Im Ergebnis wird eine angepaßte Verarbeitung, wie beispielsweise eine Bestimmung, möglich. 35

Der Hauptprozessor 204 verarbeitet vor allem im Vergleich zu einer Makroblockschicht höhere Schichten (gemäß der MPEG 1/2-Spezifikation). Mit anderen Worten, der Hauptprozessor 204 ist für Prozesse verantwortlich, die auf Grundlage der Einheit von GOP/Bild/Abschnitt ausgeführt werden. Zum Beispiel setzt der Hauptprozessor 204 die Codierparameter des nachfolgenden Bildes auf Grundlage der Menge von Codes, die beim Codieren des vorhergehenden Bildes verwendet wurden. Der Hauptprozessor 204 steuert ferner Sondermodi, wie beispielsweise Ratensteuerung, Synchronauffrischung, verstärkte Innensteuerung und so weiter. Die Hauptprozesse der Makroblockschicht umfassen eine angepaßte Quantisierung, welche auf Grundlage der berechneten Aktivität ausgeführt wird. 40 45

Wie vorstehend beschrieben, ist der Hauptprozessor 204 dazu in der Lage, auf ein Register oder einen Speicher in jeder Einheit der Steuereinheit 23 zuzugreifen, so daß er auf das Register oder den Speicher Bezug nehmen kann, wobei er einige Prozesse, wie beispielsweise eine Operation, ausführen kann und den Inhalt des Registers oder Speichers umschreiben kann, vorausgesetzt, daß diese Operationen innerhalb eines vorbestimmten annehmbaren Prozeßschritts abgeschlossen wird. Das gibt einen höheren Grad an Freiheit beim Codieren. 50

Wie aus der vorstehenden Beschreibung des Codierens erkannt werden kann, wird das Codieren in jeder der Verarbeitungseinheiten unter Pipelinesteuerung durch den Abschnitt zur Gesamtsteuerung 202 in der Steuereinheit 23 unabhängig ausgeführt. Mit anderen Worten, die vorstehend genannten vier Operationen der Datensteuerung werden unabhängig voneinander und parallel zueinander durchgeführt. Das Decodieren der Bilder kann in gleichartiger Weise wie das Codieren durch Parallelbetrieb der entsprechenden Blöcke in der Steuereinheit 23 unter der Pipelinesteuerung des Abschnitts zur Gesamtsteuerung 202 durchgeführt werden. 55

### 3. Pixelverarbeitungseinheit 60

#### (1) Pixelverarbeitungseinheit 31 (Fig. 18 und 19)

Fig. 18 stellt ein Blockschaltbild dar, welches ein Beispiel der Pixelverarbeitungseinheit zeigt. Die in Fig. 18 dargestellte Pixelverarbeitungseinheit 31 wird beispielsweise als in Fig. 1 gezeigte Pixelverarbeitungseinheit 3 verwendet. 65

Unter Bezugnahme auf Fig. 18 umfaßt die Pixelverarbeitungseinheit 31: eine Host-Schnittstellen(I/F)-Schaltung 311, eine lokale Speichergruppe 312, einen DCT/IDCT-Operator 313, einen Quantisierer 314, ein Filter 315,

einen Auswahlbus 316, eine Codedatenbus-Schnittstellenschaltung 317, eine Pixelbus-Schnittstellenschaltung 318 und eine Steuereinrichtung 319.

Die Host-Schnittstellenschaltung 311 ist zur Dateneingabe/-ausgabe in und aus dem Host-Prozessor vorgesehen, das heißt zwischen der Steuereinheit 2 und der Pixelverarbeitungseinheit 31. Die Pixelbus-Schnittstellenschaltung 318 ist zur Eingabe/Ausgabe der Bilddaten vorgesehen. Die Codedatenbus-Schnittstellenschaltung 317 ist zur Eingabe/Ausgabe der Codedaten vorgesehen.

Der DCT/IDCT-Operator 313 führt die DCT-Operation und die inverse DCT-Operation aus. Der Quantisierer 314 ist zur Quantisierung vorgesehen. Das Filter 315 ist zum Erzeugen der Referenzbilddaten zur Bewegungsvoraussage vorgesehen. Der Auswahlbus 316 ist zum Übertragen von internen Daten der Pixelverarbeitungseinheit 31 vorgesehen. Die Steuereinrichtung 319 ist zur Gesamtsteuerung der Pixelverarbeitungseinheit 31 vorgesehen.

Fig. 19 stellt ein Blockschaltbild der in Fig. 18 dargestellten Pixelverarbeitungseinheit 31 detaillierter dar. Die in Fig. 18 dargestellte lokale Speichergruppe 312 umfaßt in Bild 19 gezeigte Bildspeicher 331 bis 335, Datenspeicher 341 und 342, einen Codedatenspeicher 343 und einen Parameterspeicher 344. Der in Fig. 18 dargestellte Auswahlbus 316 umfaßt in Fig. 19 dargestellte Auswahlbusse SB1 bis SB5.

Der Bildspeicher 331 ist zum Speichern der zu komprimierenden Bilddaten vorgesehen. Die Bildspeicher 332 und 333 speichern die Referenzbilddaten von vorausgehenden und nachfolgenden Bildern. Der Bildspeicher 334 speichert die verarbeiteten Referenzbilddaten, die gefiltert worden sind. Der Bildspeicher 335 speichert die zu codierenden Bilddaten. Der Codedatenspeicher 343 speichert die quantisierten Codedaten. Der Parameterspeicher 344 speichert Tabellendaten für eine Operation, wie beispielsweise eine Quantisierungsmatrix.

Der Auswahlbus SB1 weist eine Busbreite von 12 Bit auf und überträgt die gefilterten Referenzbilddaten. Der Auswahlbus SB2 weist eine Busbreite von 12 Bit auf und überträgt die Differenzdaten nach der Differenzberechnung und nach dem inversen DCT. Der Auswahlbus SB3 weist eine Busbreite von 12 Bit auf und ist zum Übertragen des DCT-Faktors nach dem DCT und nach der inversen Quantisierung vorgesehen. Der Auswahlbus SB4 weist eine Busbreite von 12 Bit auf und ist zum Übertragen der Codedaten nach der inversen Quantisierung und nach einem Nulldurchlauf vorgesehen. Der Auswahlbus SB5 weist eine Busbreite von 16 Bit auf und ist zum Übertragen der Parameterdaten vorgesehen.

## (2) Pixelverarbeitungseinheit 32 (Fig. 20 und 21)

Fig. 20 stellt ein Blockschaltbild dar, welches ein anderes Beispiel der Pixelverarbeitungseinheit zeigt. Die in Fig. 20 dargestellte Pixelverarbeitungseinheit 32 kann ferner beispielsweise als die in Fig. 1 gezeigte Pixelverarbeitungseinheit 3 verwendet werden.

Unter Bezugnahme auf Fig. 20 umfaßt die Pixelverarbeitungseinheit 32 im Vergleich zu der in Fig. 19 dargestellten Pixelverarbeitungseinheit eine programmierbare Verarbeitungseinheit 320. Die programmierbare Verarbeitungseinheit 320 ist mit einem Auswahlbus 326 verbunden und ist zu verschiedenen anderen Operationen gemäß in einer Einheit 320 gespeicherten Programmen in der Lage.

Beispielsweise führt die programmierbare Verarbeitungseinheit 320 häufig verwendete Operationen, wie beispielsweise Addition, Subtraktion, Summation von Differenzabsolutwerten, Akkumulation von Differenzabsolutwerten, Summation von Differenzquadraten, Akkumulation von Differenzquadraten, Multiplikation und so weiter, gemäß den gespeicherten Programmen aus.

Fig. 21 stellt ein Blockschaltbild der in Fig. 20 dargestellten Pixelverarbeitungseinheit 32 detaillierter dar. Wie in Fig. 21 gezeigt, ist die hinzugefügte programmierbare Verarbeitungseinheit 320 mit den Auswahlbussen SB1 bis SB5 verbunden. Ferner ist die programmierbare Verarbeitungseinheit 320 mit dem Parameterspeicher 344 verbunden.

## (3) Betrieb der Pixelverarbeitungseinheiten 31 und 32

Unter Bezugnahme auf Fig. 19 wird der Betrieb der Pixelverarbeitungseinheit 31 beschrieben werden. Als Beispiel wird der Betrieb der Bewegtbildcodierung vom Akkumulationstyp gemäß der MPEG-Spezifikation beschrieben werden.

Die Pixelbus-Schnittstellenschaltung 318 empfängt die zu codierenden Bilddaten aus dem in Fig. 1 dargestellten Bildspeicher 51 und gibt die Daten zum Bildspeicher 331 durch den internen Pixelbus IPB2 aus. Wenn die Bewegungsvoraussage erforderlich ist, dann werden bei der Vorwärtsvoraussage die Referenzbilddaten zum Bildspeicher 332 ausgegeben. Indessen werden bei der Zwei-Weg-Voraussage die Referenzbilddaten des vorhergehenden und des darauffolgenden Bildes in die Bildspeicher 332 und 333 übertragen.

Bei der Vorwärtsvoraussage werden die im Bildspeicher 332 gespeicherten Daten zum Filter 315 ausgegeben. Ein Filtern wird durch das Filter 315 bewirkt, und die verarbeiteten Referenzdaten werden in den Bildspeicher 334 geschrieben.

Bei der Zwei-Weg-Voraussage werden die im Bildspeicher 332 gespeicherten Vorwärts-Referenzbilddaten zum Filter 315 übertragen, wogegen die im Bildspeicher 333 gespeicherten Rückwärts-Referenzbilddaten ebenfalls zum Filter 315 übertragen werden. Im Filter 315 werden eine Filterung und eine Interpolation an den übertragenen Referenzbilddaten ausgeführt und die erzeugten Referenzbilddaten in den Bildspeicher 334 geschrieben.

Ein Differenzgenerator 310 überträgt die im Bildspeicher 331 gespeicherten zu codierenden Bilddaten zum Auswahlbus SB2. Der Differenzgenerator 310 überträgt Differenzdaten zwischen den zu codierenden Bilddaten und den im Bildspeicher 334 gespeicherten Referenzbilddaten zum Auswahlbus SB2, falls erforderlich.

Der DCT/IDCT-Operator 313 führt eine DCT-Operation an den Bilddaten oder den Differenzdaten durch,

welche durch den Auswahlbus SB2 beim Codieren zugeführt werden, und überträgt den erzeugten DCT-Faktor durch den Auswahlbus SB3. Wenn der Codier- und Decodierablauf rückwärts durchgeführt wird, dann führt der DCT/IDCT-Operator 313 eine inverse DCT-Operation an durch den Auswahlbus SB3 zugeführten DCT-Faktor aus und erzeugt Bilddaten oder Differenzdaten. Die erzeugten Daten werden durch den Auswahlbus SB2 übertragen.

Der Quantisierer 314 bewirkt eine Quantisierung am durch den Auswahlbus SB3 bereitgestellten DCT-Faktor unter Bezugnahme auf den Parameterspeicher 344. Die im Ergebnis der Quantisierung erhaltenen Codedaten werden durch den Auswahlbus SB4 zum Codedatenspeicher 343 übertragen. Beim Rückwärtsablauf der Codierung und Decodierung führt der Quantisierer 314 eine inverse Quantisierung der durch den Auswahlbus SB4 bereitgestellten Codedaten aus. Der als Ergebnis der inversen Quantisierung erhaltene DCT-Faktor wird durch den Auswahlbus SB3 übertragen.

Beim Codieren empfängt die Codedatenbus-Schnittstellenschaltung 317 im Codedatenspeicher 343 gespeicherte Codedaten durch den Auswahlbus SB5. Die Codedatenbus-Schnittstellenschaltung 317 übermittelt die empfangenen Codedaten, so wie sie sind, an einen externen Codedatenbus, das heißt den in Fig. 1 dargestellten Codebus CD. In manchen Fällen werden sie durch die Codedatenbus-Schnittstellenschaltung 317 in einen Lauf-Pegel-Code umgewandelt und zum Codebus CB ausgegeben. Beim Decodieren führt die Codedatenbus-Schnittstellenschaltung 317 eine inverse Quantisierung an den durch den Auswahlbus SB4 zugeführten Codedaten aus. Der als Ergebnis der inversen Quantisierung erhaltene DCT-Faktor wird durch den Auswahlbus SB3 übertragen.

Beim Referenzbild-Decodierprozeß zum Codieren und beim Decodieren stellt ein Bilddecodierer 310 die durch den Auswahlbus SB1 zugeführten Referenzbilddaten und die Daten des Ergebnisses der Addition der decodierten Differenzdaten bereit, wenn eine Bewegungsvoraussage erforderlich ist, und die Daten werden im Bildspeicher 335 gespeichert. Wenn die Bewegungsvoraussage nicht erforderlich ist, dann werden inzwischen die decodierten Bilddaten, so wie sie sind, ausgegeben und im Bildspeicher 335 gespeichert.

Die Host-Schnittstellenschaltung 331 empfängt Operationsbefehle für die Pixelverarbeitungseinheit 31 und Parameter, wie beispielsweise die Quantisierungsmatrix, durch den in Fig. 1 dargestellten Host-Bus HB. Inzwischen gibt die Host-Schnittstellenschaltung 311 sowohl die in den internen Speichern 331 bis 335 und 341 bis 344 gespeicherten Daten als auch die in den internen Registern gespeicherten Daten nach außen aus.

#### (4) Prozeßablauf in der Pixelverarbeitungseinheit (Fig. 22 bis 25)

##### (i) Ablauf beim Codieren des durch ein Innenbild vorausgesagten Bildes gemäß der MPEG-Spezifikation

Fig. 22 stellt ein Flußdiagramm des Codierens in der in Fig. 19 gezeigten Pixelverarbeitungseinheit 31 dar. Unter Bezugnahme auf Fig. 22 werden die im Bildspeicher 331 gespeicherten Bilddaten zum Differenzgenerator 310 übertragen. Die übertragenen Daten werden in diesem Fall überhaupt nicht verarbeitet und werden dem DCT/IDCT-Operator 313 durch den Auswahlbus SB2 zugeführt. Nach der DCT-Operation im Operator 313 werden die das Ergebnis der Operation angehenden Daten an den Quantisierer 314 durch den Auswahlbus SB3 übertragen.

Der Quantisierer 314 führt eine Quantisierung unter Rückgriff auf den Parameterspeicher 344 durch den Auswahlbus SB5 aus. Die das Ergebnis der Quantisierung angehenden Daten, das heißt Codedaten, werden zum Codedatenspeicher 343 übertragen und darin gespeichert. Die im Codedatenspeicher 343 gespeicherten Daten werden dem externen Codebus CB (Fig. 1) durch den Auswahlbus SB4 und die Codedatenbus-Schnittstellenschaltung 317 zugeführt.

Fig. 23 stellt ein Flußdiagramm zum Decodieren in der in Fig. 19 gezeigten Pixelverarbeitungseinheit 31 dar. Unter Bezugnahme auf Fig. 23 werden die im Codedatenspeicher 343 gespeicherten zu decodierenden Daten zum Quantisierer 314 durch den Auswahlbus SB4 übertragen. Der Quantisierer 314 bewirkt eine inverse Quantisierung unter Rückgriff auf den Parameterspeicher 344 durch den Auswahlbus SB5. Das Ergebnis der inversen Quantisierung angehende Daten werden dem DCT/IDCT-Operator 313 durch den Auswahlbus SB3 zugeführt.

Der Operator 313 führt eine inverse DCT-Operation an den zugeführten Daten aus. Das Ergebnis der Operation angehende Daten werden dem Decodierer 310 durch den Auswahlbus SB2 zugeführt. Der Decodierer 310 führt eine Begrenzungsoperation an den zugeführten Daten aus, und die verarbeiteten Daten werden in den Bildspeicher 335 übertragen, um darin gespeichert zu werden.

##### (ii) Ablauf beim Codieren der Zwei-Weg-Voraussage gemäß der MPEG-Spezifikation

Unter Bezugnahme auf Fig. 24 werden im Bildspeicher 332 gespeicherte Bilddaten der Vorwärtsrichtung und im Bildspeicher 333 gespeicherte Bilddaten der Rückwärtsrichtung zum Filter 315 übertragen. Das Filter 315 führt einen Prozeß zum Erzeugen eines 1/2-Präzisionsbildes und eine Pixelinterpolation an den übertragenen Daten aus. Die verarbeiteten Daten werden in den Bildspeicher 334 übertragen und darin gespeichert. Die im Bildspeicher 334 gespeicherten Daten werden dem Differenzgenerator 310 durch den Auswahlbus SB1 zugeführt. Inzwischen werden die im Bildspeicher 331 gespeicherten zu codierenden Bilddaten ferner zum Differenzgenerator 310 übertragen. Der Differenzgenerator 310 führt eine Pixeldifferenz-Operation an den übertragenen Daten aus.

Der DCT/IDCT-Operator 313 empfängt die Differenzdaten aus dem Differenzgenerator 310 durch den Auswahlbus SB2. Der Operator 313 führt eine DCT-Operation an den empfangenen Daten aus, und die das Ergebnis der Operation angehenden Daten werden dem Quantisierer 314 durch den Auswahlbus SB3 zugeführt.

Der Quantisierer 314 führt eine Quantisierung an den zugeführten Daten unter Rückgriff auf den Parameterspeicher 344 durch den Auswahlbus SB5 aus. Die das Ergebnis der Quantisierung angegebenden Daten werden in den Codedatenspeicher 343 durch den Auswahlbus SB4 übertragen und darin gespeichert.

Unter Bezugnahme auf Fig. 25 werden beim Decodieren im Codedatenspeicher 343 gespeicherte Daten, das heißt zu decodierende Daten, durch den Auswahlbus SB4 zum Quantisierer 314 übertragen. Der Quantisierer 314 führt eine Quantisierung an den übertragenen Daten unter Rückgriff auf den Parameterspeicher 344 durch den Auswahlbus SB5 aus. Die das Ergebnis der Quantisierung angegebenden Daten werden dem DCT/IDCT-Operator 313 durch den Auswahlbus SB3 zugeführt.

Der Operator 313 führt ein inverses DCT an den zugeführten Daten aus. Die das Ergebnis der Operation angegebenden Daten werden dem Decodierer 310 durch den Auswahlbus SB2 zugeführt. Der Decodierer 310 empfängt die im Bildspeicher 334 gespeicherten Daten durch den Auswahlbus SB1. Der Decodierer 310 führt an den zugeführten Daten das Decodieren (eine Addition) und die Begrenzungsoperation der Pixel aus. Die verarbeiteten Daten werden in den Bildspeicher 335 übertragen und darin gespeichert.

#### (5) Pipelineverarbeitung (Fig. 26 bis 29)

Fig. 26 stellt ein Zeitablaufdiagramm dar, welches eine Pipelineverarbeitung zum Codieren in der in Fig. 19 dargestellten Pixelverarbeitungseinheit 31 zeigt. Unter Bezugnahme auf Fig. 26 werden in einem Operationszeitabschnitt A Schreibdaten für den Bildspeicher 331 durch den internen Pixelbus IPB2 übertragen. In einem Operationszeitabschnitt B werden Schreibdaten für den Bildspeicher 332 durch den internen Pixelbus IPB2 übertragen. In einem Operationszeitabschnitt C werden Schreibdaten für den Bildspeicher 333 durch den internen Pixelbus IPB2 übertragen.

In einem Operationszeitabschnitt D wird das Filter 315 betrieben. In einem Operationszeitabschnitt D werden ein DCT und eine Quantisierung durch den DCT/IDCT-Operator 313 und den Quantisierer 314 bewirkt. In einem Operationszeitabschnitt F werden sowohl eine inverse Quantisierung und ein inverses DCT als auch eine Decodierung durch den Operator 313, den Quantisierer 314 und den Bilddecodierer 310 ausgeführt. In einem Operationszeitabschnitt G wird eine Leseoperation aus dem Bildspeicher 335 ausgeführt. In einem Operationszeitabschnitt H werden eine Lauf-Pegel-Verdichtung und -Ausgabe durch die Codedatenbus-Schnittstellenschaltung 317 bewirkt.

Die in Fig. 26 dargestellten Prozesse werden Makroblock für Makroblock nacheinander ausgeführt. Insbesondere wird die Pipelineverarbeitung für eine Anzahl von Makroblöcken ausgeführt, obwohl in Fig. 26 die Verarbeitung der Bilddaten von nur einem Makroblock dargestellt ist. Die Pipelineverarbeitung für eine Mehrzahl von Makroblöcken wird nachstehend beschrieben werden.

Fig. 27 stellt ein Zeitablaufdiagramm dar, welches die Pipelineverarbeitung bei einer Innenbildvoraussage zeigt. Unter Bezugnahme auf Fig. 27 bezeichnet der Zusatz 0 die Datenverarbeitung des *O*ten Makroblocks, und der Zusatz 1 bezeichnet die Verarbeitung des ersten Makroblocks.

Fig. 28 stellt ein Zeitablaufdiagramm dar, welches die Pipelineverarbeitung bei der Ein-Weg-Voraussage zeigt. Fig. 29 stellt ein Leitablaufdiagramm dar, welches die Pipelineverarbeitung bei der Zwei-Weg-Voraussage zeigt.

Wie aus den Fig. 27 bis 29 zu erkennen ist, wird die Datenverarbeitung für eine Mehrzahl von Makroblöcken unter der Steuerung der Pipelineprozeßsteuerung in der Pixelverarbeitungseinheit 3 ausgeführt, wobei die Datenverarbeitung bezüglich der Verarbeitung in der Pixelverarbeitungseinheit 3 effizient ausgeführt werden kann. Mit anderen Worten, es wird eine Pixelverarbeitungseinheit vorgesehen, welche ein effizientes DCT und eine effiziente Quantisierung mit hoher Geschwindigkeit gestattet.

#### (6) Pixelverarbeitungseinheit 33

Fig. 93 stellt ein Blockschaltbild dar, welches ein weiteres Beispiel der Pixelverarbeitungseinheit zeigt. Die in Fig. 93 dargestellte Pixelverarbeitungseinheit 33 kann beispielsweise als in den Fig. 83 bis 89 und in der Fig. 91 gezeigte Pixelverarbeitungseinheit verwendet werden.

Unter Bezugnahme auf Fig. 93 unterscheidet sich die Pixelverarbeitungseinheit von der in Fig. 18 dargestellten Pixelverarbeitungseinheit 31 dadurch, daß sie zwei Sätze von Pixelbus-Schnittstellenschaltungen 318a und 318b umfaßt. Abgesehen von dieser Tatsache, ist sie die gleiche wie die in Fig. 18 gezeigte Pixelverarbeitungseinheit 31. Daher sind die entsprechenden Abschnitte durch die gleichen Bezugszeichen bezeichnet, und die Beschreibung derselben wird nicht wiederholt.

Fig. 94 stellt ein Blockschaltbild der in Fig. 93 gezeigten Pixelverarbeitungseinheit 33 detaillierter dar. Unter Bezugnahme auf Fig. 94 weisen die Pixelbusse IPB2a und IPB2b jeweils eine Busbreite von 32 Bit auf und sind mit zugeordneten Pixelbus-Schnittstellenschaltungen 318a und 318b und Bildspeichern 331 bis 333 und 335 entsprechend verbunden. Von diesen Tatsachen abgesehen, ist sie die gleiche wie die in Fig. 19 dargestellte Pixelverarbeitungseinheit 31. Daher sind die entsprechenden Abschnitte durch die gleichen Bezugszeichen bezeichnet, und die Beschreibung derselben wird nicht wiederholt.

Der Betrieb der Verarbeitungseinheit 33 wird beschrieben werden. Unter Bezugnahme auf Fig. 94 wird eine Bewegtbild-Codierung gemäß der MPEG-Spezifikation als ein Betriebsbeispiel beschrieben werden.

Bei der MPEG-Spezifikation werden die Vorwärtsvoraussage, die Rückwärtsvoraussage und die Zwei-Weg-Voraussage verwendet, und die Voraussagedaten müssen von außen gemäß dem Typ der Voraussage übertragen werden. Die Übertragung der Daten für die Voraussage wird durch die Pixelbus-Schnittstellenschaltung 318a oder 318b ausgeführt. Die zu codierenden Bilddaten werden ebenfalls durch die Pixelbus-Schnittstellenschaltung 318a oder 318b übertragen. Die Bildkompression wird in einem Filter 325, in der DCT/IDCT-Vorrich-

tung 313, im Quantisierer 314, im Differenzgenerator und im Bilddecodierer 310 ausgeführt, und die Daten werden durch die Codedatenbus-Schnittstellenschaltung 317 bereitgestellt. Indessen wird das in der Pixelverarbeitungseinheit 33 lokal decodierte Bild zu separaten externen Vorrichtungen durch die Pixelbus-Schnittstellenschaltungen 318a und 318b übertragen. Ein Teil der lokal decodierten übertragenen Daten dienen beispielsweise als Eingangsdaten für die in Fig. 87 dargestellte Einheit zur Bewegungsvoraussage 47. Die anderen der übertragenen lokal decodierten Daten dienen als vorausgesagte Bilddaten für die nächste Bildverarbeitung.

Wie vorstehend beschrieben, gestattet die in Fig. 94 dargestellte Pixelverarbeitungseinheit 33 die gleichzeitige Übertragung der lokal decodierten Bilddaten zu einer Mehrzahl von externen Vorrichtungen, und sie gestattet ferner die Übertragung vorausgesagter Bilddaten aus einem Port, während die Referenzbilddaten aus der Einheit zur Bewegungsvoraussage in den lokalen Speicher durch das andere Port übertragen werden. Das Übertragungsport des vorausgesagten Bildes ist nicht auf ein Port beschränkt, sondern das vorausgesagte Bild kann durch ein beliebiges anderes Port übertragen werden. Daher können der Systemaufbau und das Verfahren der Datenübertragung leicht geändert werden.

Fig. 95 ist ein Blockschaltbild, welches ein anderes Beispiel der in Fig. 93 gezeigten Pixelverarbeitungseinheit 33 darstellt. Die in Fig. 95 dargestellte Pixelverarbeitungseinheit 33a kann ferner als in den Fig. 83 bis 89 und 91 dargestellte Pixelverarbeitungseinheit verwendet werden.

Unter Bezugnahme auf Fig. 95 ist im Vergleich mit der in Fig. 94 dargestellten Pixelverarbeitungseinheit 33 in der Pixelverarbeitungseinheit 33a der Pixelbus IPB2 mit den beiden Pixelbus-Schnittstellenschaltungen 318a und 318b verbunden, und ein interner Bus wird gemeinsam für die Pixeldatenübertragung verwendet. Abgesehen von diesen Tatsachen, ist sie die gleiche wie die in Fig. 94 gezeigte Pixelverarbeitungseinheit 33. Daher wird die Beschreibung nicht wiederholt.

Der Betrieb der Pixelverarbeitungseinheit 33a wird im folgenden beschrieben werden. Unter Bezugnahme auf Fig. 95 wird der interne Pixelbus IPB2 gemeinsam verwendet. Wenn daher die lokal decodierten Daten zu den beiden Pixelbus-Schnittstellenschaltungen 318a und 318b gleichzeitig zu übertragen sind, dann wird ein gleichartiger Betrieb wie bei der in Fig. 94 dargestellten Pixelverarbeitungseinheit 32 ausgeführt.

Durch den vorstehend beschriebenen Aufbau kann die Pixelverarbeitungseinheit 33a die lokal decodierten Daten zu einer Mehrzahl von externen Vorrichtungen gleichzeitig übertragen. Während die Referenzbilddaten aus dem lokalen Speicher in die Einheit zur Bewegungsvoraussage übertragen werden, können die Daten ferner durch ein anderes Port zur Pixelverarbeitungseinheit 35 übertragen werden. Da der Pixelbus IPB2 aufgeteilt ist, kann ferner die Schaltungsfläche verkleinert werden, und sie ist daher für einen höheren Integrationsgrad geeignet.

#### 4. Prozessor für variable Längen

Nachstehend werden zwei Systemaufbauten vorgeschlagen und die Prozessoren zum Implementieren der entsprechenden Systemaufbauten beschrieben.

##### (1) Erster Systemaufbau Fig. 30 bis 35)

Fig. 30 stellt ein Blockschaltbild dar, welches einen ersten Systemaufbau zur Verarbeitung einer variablen Länge zeigt. Der Prozessor für variable Längen mit dem in Fig. 30 dargestellten Systemaufbau kann beispielsweise für den in Fig. 9 gezeigten Prozessor für variable Längen 12 verwendet werden.

Unter Bezugnahme auf Fig. 30 umfaßt der erste Systemaufbau: einen Tabellenspeicher 401 zum Codieren mit variabler Länge, einen Tabellenspeicher 402 zum Decodieren mit variabler Länge, einen Eingangsspeicher für eine Codefolge variabler Länge 403, einen Eingangsspeicher für eine Codefolge variabler Länge 404, einen Eingangsspeicher 405 für zu codierende Daten und einen Ausgangsspeicher 406 für zu codierende Daten. Der Prozessor für variable Längen 121 ist mit Speichern 401 bis 406 durch eine Busleitung BU verbunden.

Fig. 31 stellt ein Flußdiagramm des Codierens mit variabler Länge beim ersten Systemaufbau dar. Unter Bezugnahme auf Fig. 31 wird der Prozeß zum Codieren mit variabler Länge beschrieben werden. In einem Schritt 431 werden die zu codierenden Daten, welche Gegenstand der Umwandlung sind, aus dem Eingangsspeicher 405 für die zu codierende Daten in einen internen Speicher im Prozessor für variable Längen 121 ausgegeben. Dabei werden die zu codierenden Daten durch die Kombination von "RUN" und "LEVEL" festgelegt. In der folgenden Beschreibung werden die zu codierenden Daten als (RN, LV) dargestellt. Die ausgegebenen zu codierenden Daten werden einem Adreßgenerator (nicht dargestellt) im Prozessor für variable Längen 121 zugeführt, und unter Rückgriff auf den Tabellenspeicher 401 zur Codierung mit variabler Länge werden ein Code variabler Länge und dessen Codelänge erhalten. Der Code variabler Länge und die Codelänge werden im internen Speicher im Prozessor für variable Längen 121 gespeichert.

In einem Schritt 433 wird bestimmt, ob der erhaltene Code variabler Länge ein Escape-Code (ESCAPE) ist oder nicht. Wenn ein Escape-Code erhalten wird, dann wird in einem Schritt 434 ein Code von festgelegter Länge durch Koppeln des Ausgabecodes mit den Daten (RN, LV) erzeugt. Nach der Erzeugung des Codes festgelegter Länge schreitet der Ablauf zu einem Schritt 435 fort. Wenn inzwischen der Escape-Code, das heißt der Code "000001", im Schritt 433 nicht erhalten wird, dann schreitet der Ablauf zum Schritt 435 fort.

Im Schritt 435 wird durch Koppeln der Daten der Codes von variabler Länge eine Codefolge von variabler Länge erzeugt.

Nach der Erzeugung der Codefolge von variabler Länge wird die Codefolge von variabler Länge in einem Schritt 436 ausgegeben.

Fig. 32 stellt ein Flußdiagramm der Decodierung mit variabler Länge im ersten Systemaufbau dar. Die Prozesse zur Decodierung mit variabler Länge werden unter Bezugnahme auf Fig. 32 beschrieben werden.



Zuerst wird in einem Schritt 441 eine Codefolge mit variabler Länge aus dem in Fig. 30 dargestellten Eingangsspeicher für eine Codefolge variabler Länge 403 in den Prozessor für variable Längen 121 übertragen. Der Prozessor für variable Längen 121 gewinnt den Kopf des Codes variabler Länge in einem Schritt 442. Insbesondere wird auf Grundlage der Codelänge des unmittelbar vorhergehenden Codes variabler Länge der Kopf des nächsten Codes variabler Länge erkannt, und aus dem Kopf des nächsten Codes variabler Länge werden Daten mit 14 Bit gewonnen, das heißt die maximale Codelänge des Codes variabler Länge.

Anschließend werden in einem Schritt 443 unter Rückgriff auf den Tabellenspeicher 402 zum Decodieren ein Code variabler Länge und eine Codelänge, das heißt Daten (RN, LV), erhalten. Insbesondere werden Daten von 14 Bit, was die maximale Länge des vom Kopf des Codes variabler Länge beginnenden Codes variabler Länge ist, an einen nicht dargestellten Adreßgenerator ausgegeben und wird ein Adreßsignal zur Bezugnahme auf den Tabellenspeicher 402 für das Decodieren durch den Adreßgenerator erzeugt. Unter Verwendung des Adreßsignals wird auf den Tabellenspeicher 402 für das Decodieren Bezug genommen, wodurch der Code variabler Länge und seine Codelänge erhalten werden.

In einem Schritt 444 wird bestimmt, ob der Escape-Code "000001" erhalten wird oder nicht. Wenn der Escape-Code erhalten wird, dann werden in einem Schritt 445 16 Bit von den dem Escape-Code "000001" folgenden Daten als decodierte Daten (RN, LV) gewonnen. Nach dem Prozeß des Schritts 445 schreitet der Ablauf zu einem Schritt 446 fort. Wenn der Escape-Code im Schritt 444 nicht erhalten wird, dann schreitet der Ablauf zu einem Schritt 446 fort.

Im Schritt 446 werden die decodierten Daten (RN, LV) ausgegeben.

Der Datenspeicheraufbau der im Tabellenspeicher 401 zur Codierung mit variabler Länge gespeicherten Daten und der im Tabellenspeicher 402, der in Fig. 30 dargestellt ist, zur Decodierung mit variabler Länge gespeicherten Daten wird beschrieben werden. Fig. 33 zeigt den Datenspeicheraufbau des Tabellenspeichers 401 für die Codierung mit variabler Länge. Wie in Fig. 33 zu erkennen ist, sieht der Tabellenspeicher 401 für die Codierung mit variabler Länge einen gespeicherten Code variabler Länge VC und eine Codelänge CL unter Bezugnahme auf die Daten (RN, LV) vor. Die Laufdaten RN weisen eine Bitlänge von 6 Bit auf. Die Pegeldaten weisen eine Bitlänge von 10 Bit auf. Der Code variabler Länge VC hat eine Bitlänge von 14 Bit. Die Codelängendaten weisen eine Bitlänge von 4 Bit auf.

Fig. 34 stellt den Datenspeicheraufbau des Tabellenspeichers 402 für das Decodieren mit variabler Länge dar. Wie in Fig. 34 zu erkennen ist, sieht der Tabellenspeicher 402 für die Decodierung mit variabler Länge die Pegeldaten LV, die Laufdaten RN und den Code variabler Länge VC vor. Die Codelängendaten weisen eine Bitlänge von 4 Bit auf.

Die für die Bezugnahme auf den Tabellenspeicher 402 für das Decodieren mit variabler Länge verwendeten Eingangsdaten VC weisen die gleiche Bitlänge wie die maximale Codelänge des Codes variabler Länge auf. Wenn daher ein im Vergleich zur maximalen Codelänge kürzerer Code in den oberen Bits der Eingangsdaten VC vorhanden ist, dann wird das Decodierergebnis der höheren Bits mit Priorität ausgegeben, unabhängig davon, welches Bitmuster die unteren Bits aufweisen. Zum Beispiel würden die gleichen Daten 416 und 418 erhalten werden, unabhängig davon, welche der in Fig. 34 dargestellten Eingangsdaten 415 und 417 zum Bezug auf den Tabellenspeicher 402 verwendet werden.

Ein Verfahren zum Erzeugen der Codefolge von variabler Länge wird beschrieben werden. Fig. 35 stellt ein Flußdiagramm dar, welches einen Prozeß zum Erzeugen der Codefolge von variabler Länge zeigt. Es wird vorausgesetzt, daß sechs Register 421 bis 426 zum Erzeugen einer Codefolge von variabler Länge vorbereitet sind. Ferner wird vorausgesetzt, daß der Prozessor für variable Längen eine Datenverarbeitungs-Bandbreite von  $n$  Bit aufweist. Es wird nämlich angenommen, daß jedes der Register 421 bis 426 eine Bitbreite von  $n$  Bit aufweist.

Unter Bezugnahme auf Fig. 35 wird vorausgesetzt, daß das Register 421 einen Code variabler Länge VC1 und das Register 422 eine Codelänge  $k$  des Codes variabler Länge VC1 hält. Es wird angenommen, daß das Register 423 einen Code variabler Länge VC2 und das Register 424 eine Codelänge  $j$  des Codes variabler Länge VC2 hält. Es wird ferner angenommen, daß das Register 425 einen Code variabler Länge VC3 und das Register 426 eine Codelänge  $m$  des Codes variabler Länge VC3 hält.

Zuerst wird der im Register 421 gehaltene Code variabler Länge VC1 zum Register 426 übertragen. Das Register 426 speichert die entsprechenden Bits des Codes variabler Länge VC1 in einer Anordnung, welche mit einem MSB beginnt. Mit anderen Worten, das Register 426 speichert den Code variabler Länge VC1 von  $k$  Bit ab dem MSB.

In einem Schritt 427 wird die Codelänge  $j$  des Codes variabler Länge VC2 mit  $(n - k)$  verglichen. Wenn  $j < (n - k)$  ist, dann wird der im Register 423 gehaltene Code variabler Länge VC2 zum Register 426 ausgegeben. Das Register 426 hält nach dem Code variabler Länge VC1 den Code variabler Länge VC2. Die Operation im Register 426 wird bewirkt durch Verschieben des Codes variabler Länge VC2 zur LSB-Seite um  $k$  Bit und anschließendes Akkumulieren oder Ausführen einer ODER-Operation zwischen den entsprechenden Bits der Codes variabler Länge VC1 und VC2. Als Ergebnis wird das Register 426 die gekoppelten Daten von  $(k + j)$  Bit, das heißt die Codefolge von variabler Länge  $(VC1 + VC2)$ , halten.

In diesem Stadium verbleibt noch Platz von  $(n - k - j)$  Bit im Register 426, und daher wird das Register 426 ferner zum Halten des Codes variabler Länge VC3 verwendet.

In einem Schritt 427 werden aus dem Code variabler Länge VC2 im Register 423 nur die Daten von  $(n - k)$  Bit im Register 426 gehalten, wenn  $j > (n - k)$  ist. Insbesondere wird der Code variabler Länge VC2 um  $k$  Bit verschoben und eine Akkumulation oder eine ODER-Operation zwischen den Codes variabler Länge VC1 und VC2 bewirkt. Im Ergebnis wird das Register 426 durch  $k$  Bit des Codes variabler Länge VC1 und einen Teil des  $(n - k)$  Bit aufweisenden Codes variabler Länge VC2 belegt.

In einem Schritt 428 werden die Daten ausgegeben, mit denen das Register 426 belegt worden ist. Nach der



Datenausgabe wird der im Register 423 verbliebene Code variabler Länge VC2 mit  $(j + k - n)$  Bit in das Register 426 übertragen. Das Register 426 hält die übertragenen Daten, das heißt die verbliebenen Bits des Codes variabler Länge VC2, in einer Anordnung auf der Seite vom MSB.

Auf diese Weise wird das Register 426 durch nacheinander übertragene Codes variabler Länge beladen, und  $n$  Bit der eingeladenen Daten werden nacheinander als Code folge variabler Länge ausgegeben. Die Codefolge variabler Länge wird segmentiert durch wiederholtes Ausführen einer Datenverschiebung, einer logischen Operation, durch Berechnung der Codelänge des Codes variabler Länge und durch eine Bestimmung wie bei den vorstehend beschriebenen Prozessen zur Erzeugung einer Codefolge variabler Länge.

#### (2) Prozessor für variable Längen (Zugeordnete Schaltung: Fig. 36)

Fig. 36 stellt ein Blockschaltbild dar, welches ein erstes Beispiel des Prozessors für variable Längen zeigt. Unter Bezugnahme auf Fig. 36 umfaßt der Prozessor für variable Längen 122: ein Adreßregister 451, einen ROM 452 und eine Ausgangsverarbeitungs-Schaltung 453. Diese internen Schaltungen 451, 452 und 453 werden durch zugeordnete Schaltungen gebildet.

#### (3) Prozessor für variable Längen (Digitalsignal-Prozessor: Fig. 37)

Fig. 37 stellt ein Blockschaltbild dar, welches ein zweites Beispiel des Prozessors für variable Längen zeigt. Der Prozessor für variable Längen 123 wird durch einen Digitalsignal-Prozessor gebildet. Unter Bezugnahme auf Fig. 37 umfaßt der Prozessor für variable Längen 123: eine externe Schnittstellenschaltung 461, eine ALU 462, eine Verschiebeschaltung 463, einen Speicher 464, einen Anweisungsspeicher 465 und eine Steuerschaltung 466.

#### (4) Prozessor für variable Längen (Mikroprozessor zur allgemeinen Verwendung: Fig. 38)

Fig. 38 stellt ein Blockschaltbild dar, welches ein drittes Beispiel des Prozessors für variable Längen zeigt. Der Prozessor für variable Längen 124 wird durch einen Prozessor zur allgemeinen Verwendung gebildet. Unter Bezugnahme auf Fig. 38 umfaßt der Prozessor für variable Längen 124: eine Ausführungseinheit (EU) 471, welche ein Registerfile 473 umfaßt; eine Bus-Schnittstelleneinheit 472, welche ein Verlegungs-Registerfile 474 umfaßt; eine ALU 475; ein Flagregister 476; eine Bus-Schnittstelleneinheit 477; eine Anweisungskette 478 und eine Steuer-/Timingschaltung 479.

Die in den Fig. 36, 37 und 38 dargestellten entsprechenden Prozessoren für variable Längen 122, 123 und 124 weisen jeweils den in Fig. 30 gezeigten ersten Systemaufbau auf, und die vorstehend beschriebenen Prozesse des ersten Systemaufbaus werden ausgeführt.

#### (5) Zweiter Systemaufbau (Fig. 39)

Fig. 39 stellt ein Blockschaltbild dar, welches einen zweiten Systemaufbau zur Verarbeitung variabler Längen zeigt. Der in Fig. 39 dargestellte Systemaufbau ist beispielsweise in der in Fig. 1 gezeigten Vorrichtung zur Bildkompression/-expansion ausgeführt.

Unter Bezugnahme auf Fig. 39 umfaßt ein Arbeitsspeicher 7 einen Tabellenspeicher 481 für die Codierung mit variablen Längen und einen Tabellenspeicher 482 für die Decodierung mit variablen Längen. Diese Tabellenspeicher 481 und 482 sind mit dem Prozessor für variable Längen 125 durch Busse 491 und 492 verbunden.

Ein Pufferspeicher 6 umfaßt einen Eingangsspeicher 483 für eine Codefolge von variabler Länge, einen Ausgangsspeicher 484 für eine Codefolge variabler Länge, einen Eingangsspeicher 485 für zu codierende Daten und einen Ausgangsspeicher 486 für zu codierende Daten. Der Eingangsspeicher 483 und der Ausgangsspeicher 484 für die Codefolgen sind mit einem Prozessor für variable Längen 125 durch Busse 493 und 495 verbunden. Der Eingangsspeicher und der Ausgangsspeicher für die Codefolgen sind mit dem Prozessor für variable Längen 125 und einer Host-Schnittstellenschaltung 1 durch einen Bus 494 verbunden.

Der Eingangsspeicher 485 für die zu codierenden Daten und der Ausgangsspeicher 486 für die zu codierenden Daten sind mit dem Prozessor für variable Längen 125 durch Busse 496 und 498 verbunden. Der Eingangsspeicher 485 und der Ausgangsspeicher 486 sind mit dem Prozessor für variable Längen 125 und einer Pixelverarbeitungseinheit 3 durch einen Bus 497 verbunden.

#### (6) Prozessor für variable Längen (Fig. 40)

Fig. 40 stellt ein Blockschaltbild des in Fig. 39 gezeigten Prozessors für variable Längen 125 dar. Der in Fig. 40 gezeigte Prozessor für variable Längen 125 ist daher beispielsweise als in Fig. 9 dargestellter Prozessor für variable Längen 120 verwendbar.

Unter Bezugnahme auf Fig. 40 umfaßt der Prozessor für variable Längen 125: eine Steuereinrichtung 501, einen Adreßgenerator 502, eine externe Schnittstellenschaltung 503, einen Datenoperator 504, eine Schaltung zum Erzeugen/Trennen einer Codefolge variabler Länge 505, eine externe Schnittstellenschaltung 506, externe Schnittstellenschaltungen 507 und einen Datenspeicher 508. Diese internen Schaltungen 501 bis 508 sind durch einen ersten internen Bus 509 verbunden. Indessen sind die externe Schnittstellenschaltung 503 und die Schaltung zum Erzeugen/Trennen einer Codefolge variabler Länge 505 durch einen zweiten internen Bus 510 verbunden.

Die externe Schnittstellenschaltung 503 ist mit dem in Fig. 9 gezeigten Hauptport 17 durch Busse 491 und 492

verbunden. Die externe Schnittstellenschaltung 506 ist mit dem in Fig. 9 dargestellten Codedatenport 20 durch Busse 493, 494 und 495 verbunden.

(i) Adreßgenerator (Fig. 41)

Fig. 41 stellt ein Blockschaltbild des in Fig. 40 gezeigten Adreßgenerators 502 dar. Unter Bezugnahme auf Fig. 41 umfaßt der Adreßgenerator 502: ein Register für einen positiven Grenzwert 520, ein Register für einen negativen Grenzwert 521, ein PE-Modus-Register 522, ein Seitenregister 523, Adreßregister 524 und 525, einen Escape-Decodierer 526, einen Escape-Codierer 527, ein Escape-Register 528, ein Datenregister 529, eine Auswahlerschaltung 530, eine Begrenzerschaltung 531, einen Prioritäts-Codierer 532, eine Schiebeschaltung 533, ein externes Adreßregister 534, Pipelineregister 561, 562, 563 und eine ODER-Schaltung (Schaltung zum Bereitstellen einer logischen Summe) 564. Die Register 520 bis 525 sind mit einem ersten internen Bus 509 verbunden. Die Auswahlerschaltung 530 ist mit den ersten und den zweiten internen Bussen 509 und 510 verbunden. Das Datenregister 529 und das externe Adreßregister 534 sind mit der in Fig. 40 gezeigten externen Schnittstellenschaltung 503 verbunden.

Fig. 42 stellt ein Flußdiagramm des Codierens mit variabler Länge im in Fig. 41 gezeigten Adreßgenerator 502 dar. Die Prozesse im Adreßgenerator 502 werden nachstehend detailliert beschrieben werden, und nur der Grundbetrieb wird im folgenden beschrieben werden.

Unter Bezugnahme auf Fig. 42 werden in einem Schritt 601 vorgeschriebene Daten in die in Fig. 41 dargestellten Register 520 bis 523 aufgenommen. In einem Schritt 602 werden die zu codierenden Daten (RN, LV) in das Adreßregister 525 eingegeben. Die zu codierenden Daten (RN, LV) weisen insgesamt 16 Bit auf, von denen die höheren 10 Bit die Pegeldaten LV und die unteren 6 Bit die Laufdaten RN festlegen. Nach dem Schritt 602 werden Schritte 603 bis 605 und 606 bis 607 parallel ausgeführt.

Im Schritt 603 werden die im Adreßregister 525 enthaltenen Daten an die Begrenzerschaltung 531 übergeben. Die Begrenzerschaltung 531 begrenzt die höheren 10 Bit der übergebenen Daten, das heißt die Pegeldaten LV, auf einen Bereich, der durch das Register für einen positiven Grenzwert 520 und durch das Register für einen negativen Grenzwert 521 festgelegt wird. Im Ergebnis der Begrenzung werden die zu codierenden Daten mit insgesamt 12 Bit, das heißt 6 Bit der Pegeldaten LV und 6 Bit der Laufdaten RN, erhalten.

Die zu codierenden Daten mit 12 Bit werden zur ODER-Schaltung 564 übertragen, wobei eine ODER-Operation zwischen den übertragenen Daten und den aus dem Seitenregister 523 übertragenen Daten ausgeführt wird und gekoppelte Daten von 20 Bit erzeugt werden (Schritt 604). Die erzeugten Daten werden im externen Adreßregister 534 gespeichert (Schritt 605).

Im Schritt 606 werden die zu codierenden Daten im Register 525 zum Escape-Codierer 527 übertragen, und durch den Escape-Codierer 527 wird ein Code von festgelegter Länge erzeugt. Insbesondere findet ein Prozeß zum Erzeugen eines Codes von festgelegter Länge statt, wenn die zu codierenden Daten als Code von festgelegter Länge codiert werden. Im Schritt 607 wird der erzeugte Code von festgelegter Länge im Escape-Register 528 gespeichert.

Fig. 43 stellt ein Flußdiagramm des Decodierens mit variabler Länge im in Fig. 41 gezeigten Adreßgenerator 502 dar. Die Grundprozesse beim Decodieren mit variabler Länge werden beschrieben werden.

Unter Bezugnahme auf Fig. 43 werden in einem Schritt 611 vorgeschriebene Daten in die Register 520 bis 523 aufgenommen. In einem Schritt 612 wird die zu codierende Codefolge von variabler Länge in das Adreßregister 524 eingegeben. Das Register 524 speichert 22 Bit der eingegebenen Codefolge variabler Länge, welche bei der Anordnung mit dem ersten Bit auf der Seite des MSB beginnt. Nach dem Schritt 612 werden Schritte 613 bis 616 und 617 und 618 parallel ausgeführt.

Im Schritt 613 werden die im Adreßregister 524 gehaltenen Daten, das heißt die Codefolge variabler Länge, an den Prioritäts-Codierer 532 übergeben. Der Prioritäts-Codierer 532 berechnet eine Anzahl SN fortlaufender "0"en aus dem MSB der übergebenen Daten.

Im Schritt 614 werden die Daten im Adreßregister 524 zur Verschiebeschaltung 533 übertragen. In Reaktion auf die im Schritt 613 berechnete Anzahl SN gewinnt die Verschiebeschaltung 533 6 Bit der Daten, was weniger als die (SN+1)te Anzahl aus dem MSB der übertragenen Daten ist. Die gewonnenen Daten werden der ODER-Schaltung 564 zugeführt.

Im Schritt 615 werden die logische Summe durch die ODER-Schaltung 564 bereitgestellt und 20 Bit Daten erzeugt. Die erzeugten Daten werden im externen Adreßregister 534 gespeichert (Schritt 616).

Indessen werden im Schritt 617 die im Adreßregister 524 aufbewahrten Daten zum Escape-Decodierer 526 übertragen. Der Escape-Decodierer 526 führt das Decodieren aus, wobei berücksichtigt wird, daß die übertragenen Daten einen Code von festgelegter Länge aufweisen, welche den Escape-Code verwenden. Die decodierten Daten werden im Escape-Register 528 gespeichert (Schritt 618).

(ii) Codefolge von variabler Länge Erzeuge- /Trennschaltung (Fig. 44)

Fig. 44 stellt ein Blockschaltbild der in Fig. 40 gezeigten Schaltung zum Erzeugen/Trennen einer Codefolge von variabler Länge 505 dar. Unter Bezugnahme auf Fig. 44 umfaßt die Schaltung zum Erzeugen/Trennen einer Codefolge von variabler Länge 505: ein Busregister 536, eine Abschnittverschiebeeinrichtung (BSFT) 537, einen Addierer 538, ein Codelängen-Register 539, ein Codeausgangs-Register 540, ein Codeeingangs-Register 541, FIFO-Schaltungen 542 und 543 und eine Steuereinrichtung 544. Das Busregister 536 ist mit einem ersten internen Bus 509 verbunden. Der Addierer 538 ist mit einem zweiten internen Bus 510 verbunden. Die FIFO-Schaltungen 542 und 543 sind mit einer in Fig. 47 dargestellten externen Schnittstellenschaltung 506 (welche nachstehend beschrieben werden wird) durch Signalleitungen 42a bis 43c verbunden.

Fig. 45 stellt ein Flußdiagramm des Codierens mit variabler Länge in der in Fig. 44 gezeigten Schaltung zum Erzeugen/Trennen einer Codefolge von variabler Länge 505 dar. Unter Bezugnahme auf Fig. 45 wird der Grundbetrieb des Codierens mit variabler Länge beschrieben werden.

In einem Schritt 621 wird ein Code variabler Länge VC zum Busregister 536 durch den internen Bus 509 ausgegeben. Die Codelängen-Daten CL des Codes variabler Länge VC werden zum Addierer 538 durch den zweiten internen Bus 510 übertragen.

In einem Schritt 622 koppelt die Abschnittverschiebe-Einrichtung 537 einen alten Code variabler Länge VC', welcher bereits eingegeben worden ist, mit dem gerade übertragenen Code variabler Länge VC und erzeugt eine Codefolge variabler Länge VCT. In einem Schritt 623 addiert der Addierer 538 die Codelänge CL des neuen Codes von variabler Länge zu einer Codelängensumme CLS, welche die akkumulierte Summe ist.

In einem Schritt 624 wird bestimmt, ob die Codelängensumme CLS 8 Bit überschreitet oder nicht. Wenn  $CLS \geq 8$  ist, dann schreitet der Ablauf zu einem Schritt 625 fort.

Im Schritt 625 werden 8 Bit der Daten aus dem ersten Bit der Codefolge variabler Länge VCT zur FIFO-Schaltung 542 übertragen und darin gespeichert. In einem Schritt 626 wird eine neue Codelänge durch Subtraktion berechnet, das heißt  $CLS = CLS - 8$ . Nach dem Schritt 626 kehrt der Ablauf zum Schritt 624 zurück.

Im Schritt 624 wird diese Codieroperation abgeschlossen, wenn  $CLS < 8$  ist.

Fig. 46 stellt ein Flußdiagramm des Decodierens mit variabler Länge in der in Fig. 44 gezeigten Schaltung zum Erzeugen/Trennen einer Codefolge von variabler Länge 505 dar. Unter Bezugnahme auf Fig. 46 wird in einem Schritt 631 eine Codefolge variabler Länge VCT an den ersten internen Bus 509 gelegt. In einem Schritt 632 wird die in der Codefolge variabler Länge VCT enthaltene Codelänge CL des ersten Codes von variabler Länge VC zum Addierer durch den zweiten internen Bus 510 übertragen. In einem Schritt 633 addiert der Addierer 538 die neue Codelänge CL zur Summe der alten Codelängen CLS.

In einem Schritt 634 wird bestimmt, ob die Codelängensumme CLS 8 Bit überschreitet oder nicht. Wenn  $CLS > 8$  ist, dann schreitet der Ablauf zu einem Schritt 635 fort.

Im Schritt 635 wird ein Wort der Daten aus dem Eingangsspeicher der Codefolge variabler Länge 483 übertragen, welches mit der vorliegenden Codefolge von variabler Länge gekoppelt wird. In einem Schritt 636 werden die ersten 8 Bit Daten von der Codefolge variabler Länge abgeschnitten. In einem Schritt 637 wird durch Ausführen von  $CLS = CLS - 8$  eine neue Codelängensumme CLS erhalten. Nach dem Schritt 637 kehrt der Ablauf zum Schritt 634 zurück.

Im Schritt 634 wird der Decodierprozeß abgeschlossen, wenn  $CLS < 8$  ist.

### (iii) Externe Schnittstellenschaltung 506 (Fig. 47)

Fig. 47 stellt ein Blockschaltbild der in Fig. 40 gezeigten externen Schnittstellenschaltung 506 dar. Unter Bezugnahme auf Fig. 47 umfaßt die externe Schnittstellenschaltung 506: eine Steuereinrichtung 545 mit der Funktion eines DMA, Lesezugriffszähler 546 und 549, Schreibzugriffszähler 547 und 548, Subtrahierer 565 und 566, Vergleichseinrichtungen 550 und 558, Flagerzeugungsregister 551 bis, 554, ein Schreibdatenregister 556, ein Lesedatenregister 557 und eine Auswähleinrichtung 555.

Die Steuereinrichtung 545 ist mit den in Fig. 44 dargestellten FIFO-Schaltungen 542 und 543 durch Signalleitungen 42b, 42c, 43b und 43c verbunden. Das Register 556 ist mit der in Fig. 44 gezeigten FIFO-Schaltung 542 durch eine Signalleitung 42a verbunden. Das Register 557 ist mit der in Fig. 44 gezeigten FIFO-Schaltung 543 durch eine Signalleitung 43a verbunden. Die Zähler 546 und 547 sind mit dem Eingangsspeicher für eine Codefolge von variabler Länge 438 und dem Ausgangsspeicher für eine Codefolge von variabler Länge 484, welcher in Fig. 39 dargestellt ist, durch eine Signalleitung 493 verbunden. Die Zähler 548 und 549 sind mit dem Eingangsspeicher für eine Codefolge von variabler Länge 438, dem Ausgangsspeicher für eine Codefolge von variabler Länge 484 und einer Host-Schnittstellenschaltung 1 durch eine Signalleitung 494 verbunden. Die Auswähleinrichtung 555 ist mit dem Eingangsspeicher für eine Codefolge von variabler Länge 438 und dem Ausgangsspeicher für eine Codefolge von variabler Länge 484, welcher in Fig. 39 dargestellt ist, durch eine Signalleitung 495 verbunden.

Der Zähler 546 zählt die Anzahl von Lesezugriffen auf den in Fig. 39 dargestellten Eingangsspeicher für eine Codefolge von variabler Länge 483. Der Zähler 547 zählt die Anzahl von Schreibzugriffen auf den in Fig. 39 dargestellten Ausgangsspeicher für eine Codefolge von variabler Länge 484. Der Zähler 548 zählt die Anzahl von Schreibzugriffen auf den Eingangsspeicher für eine Codefolge von variabler Länge 483 aus anderen Schaltungen. Der Zähler 549 zählt die Anzahl von Lesezugriffen auf den Ausgangsspeicher für eine Codefolge von variabler Länge 484 aus anderen Schaltungen.

Das Flagregister 551 erzeugt ein Voll-Flag, welches den Vollzustand des Eingangsspeichers für eine Codefolge von variabler Länge 483 anzeigt, und hält das Flag. Das Register 552 erzeugt und hält ein Leer-Flag, welches auf den Leerzustand des Eingangsspeichers für eine Codefolge von variabler Länge 483 hinweist. Das Flagregister 553 erzeugt und hält ein Voll-Flag, welches auf den Vollzustand des Ausgangsspeichers für eine Codefolge von variabler Länge 484 hinweist. Der Zähler 554 erzeugt und hält ein Leer-Flag, welches auf den Leerzustand des Ausgangsspeichers für eine Codefolge von variabler Länge 484 hinweist.

Die Vergleichseinrichtung 550 stellt ein Voll-Flag FF1, das auf den Vollzustand hinweist, oder ein Leer-Flag EF1, das auf den Leerzustand des Eingangsspeichers für eine Codefolge von variabler Länge 483 hinweist. Die Vergleichseinrichtung 558 stellt ein Voll-Flag FF2, das auf den Vollzustand hinweist, oder ein Leer-Flag EF2, das auf den Leerzustand des Ausgangsspeichers für eine Codefolge von variabler Länge 484 hinweist.

Fig. 48 stellt ein Flußdiagramm des Codierens mit variabler Länge in der in Fig. 47 gezeigten externen Schnittstellenschaltung 506 dar. Unter Bezugnahme auf Fig. 48 wird der Grundbetrieb zum Codieren mit variabler Länge beschrieben werden.

In einem Schritt 641 wird ermittelt, ob der Ausgangsspeicher für eine Codefolge von variabler Länge 484 leer ist oder nicht. Mit anderen Worten, es wird ermittelt, ob von der Vergleichseinrichtung 558 ein Leer-Flag EF2 vorgesehen wird oder nicht. Wenn das Leer-Flag EF2 nicht ermittelt wird, dann schreitet der Ablauf zu einem Schritt 642 fort. Wenn das Leer-Flag EF2 ermittelt wird, dann ist der Prozeß beendet.

5 Im Schritt 642 wird ermittelt, ob der Ausgangsspeicher für eine Codefolge von variabler Länge 484 voll ist oder nicht. Mit anderen Worten, es wird ermittelt, ob von der Vergleichseinrichtung 558 ein Voll-Flag EF2 vorgesehen wird oder nicht. Wenn das Voll-Flag EF2 nicht ermittelt wird, dann schreitet der Ablauf zu einem Schritt 643 fort. Wenn das Voll-Flag EF2 ermittelt wird, dann ist der Prozeß beendet.

10 Im Schritt 643 wird ein Wort der im Ausgangsspeicher für eine Codefolge von variabler Länge 484 gespeicherten Daten gelesen und im Ausgangsspeicher für eine Codefolge von variabler Länge gespeichert. In einem Schritt 644 wird der Schreibzugriffszähler 547 auf summiert. Im Anschluß an Schritt 644 kehrt der Ablauf zum Schritt 641 zurück.

Fig. 49 stellt ein Flußdiagramm des Decodierens mit variabler Länge in der in Fig. 47 gezeigten externen Schnittstellenschaltung 506 dar. Unter Bezugnahme auf Fig. 49 wird in einem Schritt 651 ermittelt, ob der 15 Eingangsspeicher für eine Codefolge von variabler Länge 483 voll ist oder nicht. Mit anderen Worten, es wird ermittelt, ob von der Vergleichseinrichtung 550 ein Voll-Flag EF1 vorgesehen wird oder nicht. Wenn das Voll-Flag EF1 nicht ermittelt wird, dann schreitet der Ablauf zu einem Schritt 652 fort. Wenn das Voll-Flag EF1 ermittelt wird, dann ist der Prozeß beendet.

20 Im Schritt 652 wird ermittelt, ob der Eingangsspeicher für eine Codefolge von variabler Länge 483 leer ist oder nicht. Mit anderen Worten, es wird ermittelt, ob aus der Vergleichseinrichtung 550 ein Leer-Flag EF1 vorgesehen ist oder nicht.

Wenn das Leer-Flag EF1 nicht ermittelt wird, dann schreitet der Ablauf zu einem Schritt 653 fort. Wenn das Leer-Flag EF1 ermittelt wird, dann ist der Prozeß beendet.

25 Im Schritt 643 wird ein Wort der im externen Eingangsspeicher für eine Codefolge von variabler Länge gespeicherten Daten gelesen und im Eingangsspeicher für eine Codefolge von variabler Länge 483 gespeichert. In einem Schritt 654 wird der Lesezugriffszähler 446 für den externen Eingangsspeicher für eine Codefolge von variabler Länge auf summiert. Im Anschluß an Schritt 654 kehrt der Ablauf zum Schritt 651 zurück.

#### (iv) Detaillierte Beschreibung des Codierens/Decodierens mit variabler Länge

30 In folgenden wird das Codieren/Decodieren mit variabler Länge in dem in Fig. 39 dargestellten Prozessor für variable Längen 125 detaillierter beschrieben werden.

Fig. 50 stellt eine Tabelle der Codes variabler Länge dar, welche in dem in Fig. 39 dargestellten Prozessor für variable Längen 125 verwendet wird. Unter Bezugnahme auf Fig. 50 entsprechen die vier Spalten auf der linken 35 Seite der Tabelle der Codes variabler Länge der Tabelle der Codes variabler Länge gemäß der H.261-Empfehlung des CCITT. Insbesondere sind die in der H.261 spezifizierten Laufdaten (Null-Lauflänge) RN, Pegeldaten (Faktor) LV, die Codelänge CL und der Code von variabler Länge VC in vier Spalten auf der linken Seite aufgelistet. Indessen stellen die drei Spalten auf der rechten Seite eine gruppierte Tabelle der Codes variabler Länge dar. Insbesondere sind der Code von variabler Länge VC und die Codelänge CL gemäß der Anzahl 40 fortlaufender "0"en ab dem ersten Bit des Codes von variabler Länge VC in insgesamt neun Gruppen GR0 bis GR8 eingeteilt.

Beispielsweise umfaßt die Gruppe GR0 einen Code variabler Länge, dessen erstes Bit keine "0" ist. Die Gruppe GR1 umfaßt einen Code variabler Länge VC, bei welchem nur das erste Bit "0" ist. Die Gruppe GR2 45 umfaßt einen Code variabler Länge VC, bei welchem die ersten beiden Bits "0" sind. In gleichartiger Weise umfaßt die Gruppe GR8 einen Code variabler Länge VC, welcher 8 "0"en ab dem ersten Bit des Codes von variabler Länge VC enthält. Die gruppierten Codes variabler Länge VC und die Codelängen CL werden bei den Prozessen im Prozessor für variable Längen 125 verwendet, was nachstehend beschrieben werden wird.

50 Zuerst wird das Codieren einer variablen Länge beschrieben. Grundsätzlich wird das Codieren mit variabler Länge gemäß dem in Fig. 31 dargestellten Ablauf ausgeführt, welcher bereits beschrieben worden ist. Zuerst werden die zu codierenden Daten, das heißt die Daten (RN, LV), aus dem in Fig. 39 gezeigten Eingangsspeicher 485 für zu codierende Daten in den in Fig. 40 gezeigten Datenspeicher 538 im Prozessor für variable Längen 125 eingegeben. Die Eingabedaten werden zum Adreßgenerator 502 übertragen, welcher eine Tabellenspeicher- 55 adresse zur Umwandlung der Codierung mit variabler Länge erzeugt. Parallel zur Erzeugung der Tabellenadresse erzeugt der in Fig. 41 dargestellte Escape-Decodierer 526 unter Verwendung eines Escape-Codes einen Code von festgelegter Länge. Der erzeugte Code festgelegter Länge wird übertragen und im Escape-Register 528 gehalten. Der Betrieb des Adreßgenerators 502 wird nachstehend detaillierter beschrieben werden.

Unter Verwendung der durch den Adreßgenerator 502 erzeugten Tabellenadresse wird auf den Tabellenspeicher für die Codierung mit variabler Länge 481 Bezug genommen und werden der Code von variabler Länge VC, die Codelänge CL und das gespeicherte Escape-Flag ESF gelesen. Der Code von variabler Länge VC und die 60 Codelänge CL werden zum in Fig. 41 dargestellten Datenregister 529 übertragen und darin gehalten.

Nach Beendigung des Zugriffs auf den Tabellenspeicher 481 für die Codierung mit variabler Längen werden die Daten des Escape-Registers 528 oder des Datenregisters 529 an die Schaltung zum Erzeugen/Trennen des Codes von variabler Länge 505 übergeben. Die Auswahl wird durch die Auswahlschaltung 530 bewirkt. Insbesondere 65 gibt die Auswahlleinrichtung 530 beliebige in den beiden Registern 528 und 529 gehaltene Daten in Reaktion auf das höchstwertige Bit (MSB) selektiv aus.

Wie in Fig. 51 dargestellt, speichert der Tabellenspeicher 481 für die Codierung mit variabler Länge insbesondere ein Escape-Flag ESF, welches anzeigt, ob die zu codierenden Daten im verwendbaren Bereich des Escape-Codes am höchstwertigen Bit liegen oder nicht. Wenn daher das höchstwertige Bit der in das Datenregister

übertragenen Daten, das heißt das Flag ESF, die Anwendung des Escape-Codes anzeigt, dann gibt die Auswahl-einrichtung 530 diejenigen Daten, die im Datenregister 529 gehalten worden sind, selektiv aus. Im Ergebnis kann nur durch Bezugnehmen auf den Tabellenspeicher 481 für die Codierung mit variabler Länge bestimmt werden, ob es erforderlich ist, den Escape-Code auf die zu codierenden Daten anzuwenden, und eine Bestimmung gemäß einem Programm ist für diese Bestimmung nicht notwendig.

Im in Fig. 41 dargestellten Adreßgenerator 502 können ferner der Zugriff auf den Tabellenspeicher 481 für die Codierung mit variabler Länge und die Erzeugung des Codes von festgelegter Länge unter Verwendung des Escape-Codes parallel, das heißt gleichzeitig, ausgeführt werden. Daher kann eine durch den Zugriff auf den Tabellenspeicher 481 für die Codierung mit variabler Länge hervorgerufene Zunahme der Prozeßzeit verhindert werden.

Der durch die Bezugnahme auf den Tabellenspeicher 481 für die Codierung mit variabler Länge erhaltene Code von variabler Länge VC wird zur Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 durch den in Fig. 41 gezeigten ersten internen Bus 509 übertragen. Inzwischen wird seine Codelänge CL zur Schaltung zum Erzeugen/Trennen des Codes von variabler Länge 505 durch den zweiten internen Bus 510 übertragen. Die Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 erzeugt eine Codefolge von variabler Länge VCT aus dem übertragenen Code von variabler Länge VC und seiner Codelänge CL. Der Betrieb des Erzeugens der Codefolge von variabler Länge wird nachstehend beschrieben. Die erzeugte Codefolge von variabler Länge VCT wird zu dem in Fig. 39 dargestellten Ausgangsspeicher für eine Codefolge von variabler Länge 484 durch die externe Schnittstellenschaltung 506 übertragen. Auf diese Weise wird das Codieren mit variabler Länge abgeschlossen.

Das Decodieren mit variabler Länge wird beschrieben werden. Zuerst wird eine Codefolge variabler Länge aus dem in Fig. 39 dargestellten Eingangsspeicher für eine Codefolge von variabler Länge 483 in die Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 durch die externe Schnittstellenschaltung 506 eingegeben. Die Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 bewirkt die Segmentierung des Codes von variabler Länge, welcher Gegenstand des Decodierens ist, aus der übertragenen Codefolge von variabler Länge und verschiebt die Bits des Codes von variabler Länge auf die Seite des MSB. Der verschobene Code variabler Länge wird zum Adreßgenerator 502 durch den ersten internen Bus 509 übertragen. Eine detaillierte Beschreibung der Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 wird nachstehend gegeben werden.

Der Adreßgenerator 502 erzeugt eine Tabellenadresse für eine Bezugnahme auf den in Fig. 39 dargestellten Tabellenspeicher 482 zur Decodierung mit variabler Länge in Reaktion auf den übertragenen Code von variabler Länge. Die erzeugte Tabellenadresse wird zum Tabellenspeicher 482 für die Decodierung mit variabler Länge durch die externe Schnittstellenschaltung 503 übertragen. Im Ergebnis werden der decodierte Code von variabler Länge, die Codelängendaten und das Escape-Flag ESF im in Fig. 41 dargestellten Datenregister 529 gehalten. Zu dieser Zeit wird parallel zu diesem Prozeß ein Code von variabler Länge zu einem Escape-Code-Codierer übertragen und das Decodieren des Codes von festgelegter Länge unter Verwendung des Escape-Codes ausgeführt. Die das Ergebnis anzeigenden Daten werden in das Escape-Register 528 übertragen und darin gehalten.

Wie beim Codieren mit variabler Länge speichert der Tabellenspeicher 482 für das Decodieren mit variabler Länge ein Escape-Flag ESF. Daher wird bei Beendigung des Zugriffs auf den Tabellenspeicher 482 das Escape-Flag ESF am höchstwertigen Bit (MSB) des Datenregisters 529 gehalten. In Reaktion auf das Escape-Flag ESF wird ein Teil der im Escape-Register 528 und im Datenregister 529 gehaltenen Daten durch die Auswahl-einrichtung 530 selektiv ausgegeben. Die Ausgabedaten werden in den in Fig. 40 dargestellten Datenspeicher 508 übertragen.

Inzwischen werden die Codelängendaten zur Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 durch den zweiten internen Bus 510 übertragen. Die Schaltung zum Erzeugen/Trennen der Codefolge von variabler Länge 505 bewirkt die Segmentierung des nachfolgenden Codes variabler Länge durch Verwenden der übertragenen Codelängendaten. Die dem Datenspeicher 508 zugeführten Daten, welche das Ergebnis des Decodierens angeben, werden an den Ausgangsspeicher 486 für die zu codierenden Daten durch die externe Schnittstellenschaltung 506 ausgegeben. Somit ist das Decodieren mit variabler Länge beendet.

Im folgenden wird der Betrieb der in Fig. 41 dargestellten Adreßerzeugungsschaltung 502 detaillierter beschrieben werden.

Fig. 52 zeigt ein Datenformat der beim Codieren mit variabler Länge zu codierenden Daten. Unter Bezugnahme auf Fig. 52 umfassen die zu codierenden Daten Pegeldaten LV von 10 Bit und Laufdaten RN von 6 Bit. Daher weisen die zu codierenden Daten eine Bitlänge von insgesamt 16 Bit auf.

Unter Bezugnahme auf die vier Spalten auf der linken Seite der in Fig. 50 dargestellten Tabelle für Codes von variabler Länge weisen nun die Pegeldaten LV (welche in der H.261-Empfehlung als Absolutwertdaten festgelegt sind) einen Wert auf, welcher innerhalb des Bereichs von  $-15$  bis  $+15$  liegt. Gemäß der H.261-Empfehlung sind die außerhalb dieses Bereichs liegenden Daten dazu bestimmt, im Escape-Code (ESCAPE) dargestellt zu werden. Das bedeutet, daß die zum Darstellen der Pegeldaten LV notwendige Bitzahl 5 Bit beträgt und daß eine Kombination von Laufdaten und Pegeldaten durch insgesamt 11 Bit, zusammen mit den Laufdaten RN, dargestellt werden kann.

Die Pegeldaten LV, welche nicht niedriger als  $+15$  und nicht höher als  $-15$  sind, sind auf  $+16$  und  $-16$  entsprechend beschränkt, und daher bedeuten die Pegeldaten  $+16$  und  $-16$  umfassenden Adressen Escape-Codes. Durch diese Verarbeitung genügt es, daß die Adresse insgesamt 11 Bit aufweist, um auf die Speichertabelle zum Codieren mit variabler Länge zuzugreifen, und daher kann die Speichertabelle zum Codieren mit variabler Länge innerhalb des Bereichs von  $2k$  Worten des Adreßraums implementiert werden.

Auf diese Weise kann der Adreßgenerator 502 die Adressenerzeugung durch nur einen Pfad ausführen. Daher

begrenzt die in Fig. 41 dargestellte Begrenzeinrichtung 531 10 Bit der Pegeldaten LV der im Adreßregister 525 gehaltenen Daten auf den Bereich von +16 bis -16. Daher wird ein positiver Wert +16 in das Register für einen positiven Grenzwert 520 geschrieben, wohingegen ein negativer Wert -16 in das in Fig. 41 dargestellte Register für einen negativen Grenzwert 521 geschrieben wird. Die Begrenzeinrichtung 531 führt die vorstehend genannte Begrenzungsoperation unter Bezugnahme auf die in den Registern 520 und 521 gehaltenen Daten aus.

Im Ergebnis des Begrenzens der Pegeldaten werden insgesamt 11 Bit von Kombinationsdaten erhalten, und durch Verwenden dieser Daten wird eine Adresse zur Bezugnahme auf den Tabellenspeicher 481 zum Codieren mit variabler Länge erzeugt.

Das Decodieren bei der in Fig. 41 dargestellten Adreßerzeugungsschaltung 502 wird beschrieben werden.

Wie es in der in Fig. 50 gezeigten Tabelle von Codes variabler Länge dargestellt ist, sind die Codes von variabler Länge VC in insgesamt neun Gruppen GR0 bis GR8 eingeteilt. In jeder Gruppe sind 6 Bit zum Festlegen jedes Codes variabler Länge erforderlich. Um eine Decodiertabelle für den Code von variabler Länge zu implementieren, werden daher 4 Bit zum Angeben der insgesamt neun Gruppen und 6 Bit zum Angeben des Codes in jeder Gruppe benötigt. Insbesondere werden insgesamt 10 Bit als Adresse für die Tabelle zum Decodieren des Codes variabler Länge benötigt. Die in Fig. 41 dargestellte Adreßerzeugungsschaltung 502 kann 10 Bit der Tabellenadresse durch einen Pfad erzeugen.

Insbesondere wird der zu decodierende Code von variabler Länge zum Adreßregister 524 übertragen und darin gespeichert. Die durch das Adreßregister 524 gehaltenen Daten werden zum Prioritäts-Codierer 532 übertragen, und die Anzahl der fortlaufenden "0"en ab dem ersten Bit der übertragenen Daten wird gezählt. Folglich wird bestimmt, zu welcher der in Fig. 50 gezeigten Gruppen GR0 bis GR8 die Daten gehören, und die Gruppennummer wird durch 4 Bit der Daten dargestellt. Indessen werden 6 Bit der den fortlaufenden "0"en folgenden Daten segmentiert. Die Gruppennummer-Daten von 4 Bit und die segmentierten Daten von 6 Bit bilden eine Tabellenadresse von insgesamt 10 Bit, und die Tabellenadresse wird zum Adreßregister 534 übertragen und darin gehalten.

Auf diese Weise kann durch Verwenden des in Fig. 41 dargestellten Adreßgenerators 502 der Adressenraum für den Tabellenspeicher 481 für das Codieren mit variabler Länge und den Tabellenspeicher 482 für das Decodieren mit variabler Länge verkleinert werden, und ferner können diese Prozesse durch eine Serienoperation ausgeführt werden.

Der Codierbetrieb in der in Fig. 44 dargestellten Schaltung zum Erzeugen/Trennen einer Codefolge von variabler Länge wird beschrieben werden. Das Codeausgangsregister 540 umfaßt Parallelschieberegister 40a bis 40e, von denen jedes 8 Bit enthält. Im Ausgangszustand halten das Codeausgangsregister 540, das Codelängenregister 539 und die FIFO-Schaltung 542 jeweils die Daten "0". Die Abschnittverschiebeeinrichtung 537 mit 40 Bit (40b) verschiebt die Eingangsdaten gemäß der im Codelängenregister 539 gehaltenen Codelänge zur rechten Seite.

Beispielsweise wird "001010" als codierter erster Code von variabler Länge in das Busregister 536 übertragen, während die Codelänge "6" zum Addierer 538 von 5 Bit übertragen wird. Da der Wert im Codelängenregister 539 "0" ist, gehen in diesem Stadium die Daten im Busregister 536, so wie sie sind, durch die Abschnittverschiebeeinrichtung 537 und werden dem Codelängenregister 539 zugeführt. Im Addierer 538 wird die Eingangsdatenangabe "6" zur Datenangabe "0" im Codelängenregister 539 addiert, und die sich ergebende Datenangabe "6" werden in das Codelängenregister 539 geschrieben.

Danach wird als nächster codierter Code von variabler Länge die Datenangabe "00001000" zum Busregister 536 übertragen, und die Codelänge "8" wird dem Addierer 538 übergeben. Da die Datenangabe im Codelängenregister 539 "6" lautet, werden die Daten des Busregisters 536 nach der rechten Seite um 6 Bit durch die Abschnittverschiebeeinrichtung 537 verschoben, und die verschobenen Daten werden dem Codelängenregister 539 zugeführt. Daher enthalten die im Codelängenregister 539 gehaltenen Daten den ersten Code variabler Länge in den ersten 6 Bit und den nächsten Code variabler Länge ab dem 7ten sowie den folgenden Bits. Der Addierer 538 addiert die Eingangsdatenangabe "8" zur Datenangabe "6" im Codelängenregister 539 und überträgt das Additionsergebnis, das heißt "14", zum Codelängenregister 539.

Da das Register 40a im Codeausgangsregister 540 in diesem Stadium mit dem Code variabler Länge "00101000" belegt ist, werden die Daten im Datenregister 40a an die FIFO-Schaltung 542 ausgegeben. Danach werden die Daten im Register 40b in das Register 40a übertragen. Ferner werden die Daten im Register 40c in das Register 40b übertragen. Gleichartige Operationen werden wiederholt und die Daten des letzten Registers 40e in das Register 40d übertragen. Nach diesem Übertragungsprozeß wird die Codelänge der an die FIFO-Schaltung 542 ausgegebenen Datenangabe, das heißt "8", von der Datenangabe im Codelängenregister 539 subtrahiert. Da in diesem Beispiel das Register 539 insbesondere die Datenangabe "14" hält, wird das Subtraktionsergebnis "6" (= 14 - 8) lauten.

Im Ergebnis geben die Daten im Codelängenregister 539 die Bitzahl des im Register 40a gespeicherten Codes variabler Länge "0011000" an, und wenn der nächste Code von variabler Länge eingegeben wird, dann wird der eingegebene Code von variabler Länge zur rechten Seite um die Bitzahl des im Register 40a gespeicherten Codes von variabler Länge verschoben. Der nächste Code variabler Länge, der dem bereits gespeicherten Code variabler Länge folgt, wird nämlich im Codeausgangsregister 40 gespeichert. Auf diese Weise kann durch Verwenden des Codes variabler Länge und seiner Codelänge die Codefolge von variabler Länge ohne Bestimmung gemäß einer Software erzeugt werden.

Der Decodierbetrieb in der Schaltung zum Erzeugen/Trennen einer Codefolge von variabler Länge wird beschrieben werden. Wie im Codeausgangsregister 540, umfaßt das Codeeingangsregister 541 Parallelschieberegister 41a bis 41e, welche 8 Bit enthalten. Im Ausgangszustand halten das Codeeingangsregister 541 und das Codelängenregister 539 beide die Datenangabe "0". Die Abschnittverschiebeeinrichtung 537 mit 40 Bit ver-



schiebt die Eingangsdaten nach der linken Seite um den im Codelängenregister 539 gehaltenen Wert.

Zuerst wird die erste Codefolge variabler Länge in die Register 41a bis 41e durch die FIFO-Schaltung 543 nacheinander eingegeben. Wenn der Code von variabler Länge zu segmentieren ist, dann werden die Daten im Codeeingangsregister 541 zum Busregister 536 durch die Abschnittverschiebeeinrichtung 537 übertragen.

Da die Datenangabe im Codelängenregister 539 "0" lautet, gehen in diesem Stadium die Daten im Codeeingangsregister 541, so wie sie sind, durch die Abschnittverschiebeeinrichtung 537 und werden dem Busregister 536 zugeführt. Durch Decodieren der Datenangabe im Busregister 536 wird als Codelänge derselben die Datenangabe "6" erhalten, und die Codelängendatenangabe wird dem Addierer 538 zugeführt. Der Addierer 538 addiert die Eingabedatenangabe "6" zur Datenangabe "0" im Codelängenregister 539, und die das Ergebnis anzeigende Datenangabe "6" wird zum Codelängenregister 539 übertragen und darin gehalten.

Wenn der nächste Code variabler Länge zu segmentieren ist, dann werden die Daten im Codeeingangsregister 541 zur Abschnittverschiebeeinrichtung 537 und zum Busregister 536 übertragen. Da die Datenangabe im Codelängenregister 539 "6" lautet, wird die Datenangabe im Codeeingangsregister 541 nach der linken Seite um 6 Bit durch die Abschnittverschiebeeinrichtung 537 verschoben und die verschobene Datenangabe dem Busregister 536 zugeführt. Insbesondere wird ein dem bereits codierten Code variabler Länge folgender nächster Code variabler Länge erhalten, welcher zur linken Seite verschoben ist. Als Ergebnis des Decodierens der Datenangabe im Busregister 536 wird als Codelänge derselben die Datenangabe "8" erhalten, und die Codelängen-Datenangabe wird in den Addierer 538 eingegeben.

Der Addierer 538 addiert die Eingabedatenangabe "8" zur Datenangabe "6" im Codelängenregister 539 und überträgt die die Ergebnisse anzeigende Datenangabe "14" zum Codelängenregister 539. In diesem Stadium ist der Inhalt des Registers 41a vollständig decodiert, und daher werden die Daten im Register 40b in das Register 40a übertragen. Anschließend werden die Daten im Register 40c in das Register 40b übertragen. Durch Wiederholen gleichartiger Operationen werden die Daten im Register 40e schließlich in das Register 40d übertragen. Die nächste Codefolge von variabler Länge wird aus der FIFO-Schaltung 543 in das leere Register 40e überführt.

Da das Decodieren von 8 Bit abgeschlossen worden ist, lautet die Datenangabe im Codelängenregister 539: "6". Wenn daher der nächste Code variabler Länge segmentiert wird, dann wird ein dem bereits decodierten Code variabler Länge folgender nächster Code variabler Länge erhalten, der zur linken Seite verschoben ist.

Durch Verwenden der Codelänge des decodierten Codes variabler Länge kann auf diese Weise, ohne der Bestimmung gemäß einer Software, der zu decodierende nächste Code variabler Länge segmentiert werden, welcher zur linken Seite von der Codefolge variabler Länge verschoben ist.

Das Codieren mit variabler Länge bei der in Fig. 47 dargestellten externen Schnittstellenschaltung 506 wird beschrieben werden.

Unter Bezugnahme auf Fig. 47 steuert eine Steuereinrichtung 545 mit DMA-Funktion die interne Schaltungsanordnung in der externen Schnittstellenschaltung 506. Die Zähler 548 und 549 empfangen ein Strobe-Signal des Zugriffs (Lesen und Schreiben) auf den Eingangsspeicher für eine Codefolge variabler Länge 483 und auf den Ausgangsspeicher für eine Codefolge variabler Länge 484 durch die Signalleitung 494. Daher zählt der Zähler 548 die Anzahl der Schreibzugriffe auf den Eingangsspeicher für eine Codefolge variabler Länge 483 durch den Host-Computer. Indessen zählt der Zähler 548 die Anzahl der Lesezugriffe aus der externen Schnittstellenschaltung 506 auf den Eingangsspeicher für eine Codefolge variabler Länge 483. Der Zähler 547 zählt die Anzahl der Schreibzugriffe aus der externen Schnittstellenschaltung 506 auf den Ausgangsspeicher für eine Codefolge variabler Länge 484.

Die Ausgangsdaten aus den Zählern 546 und 548 werden dem Subtrahierer 565 zugeführt, in welchem eine Subtraktion bewirkt wird. Die das Ergebnis der Subtraktionseinrichtung anzeigende Datenangabe, folglich die Anzahl von Datenworten, wird im Eingangsspeicher für eine Codefolge variabler Länge 483 gespeichert. Die aus dem Subtrahierer 565 ausgegebenen Daten werden zur Vergleichseinrichtung 550 übertragen.

Die Vergleichseinrichtung 550 empfängt die Wortzahl-Daten im Voll-Zustand des Eingangsspeichers für eine Codefolge variabler Länge 483. Inzwischen empfängt die Vergleichseinrichtung 550 die Wortzahl-Datenangabe, welche den Leer-Zustand des Eingangsspeichers für eine Codefolge variabler Länge 483 anzeigt. Die Vergleichseinrichtung 550 vergleicht die Ausgangsdaten aus dem Subtrahierer 565 mit den Daten aus den Registern 551 und 552 und setzt Flags FF1 und EF1.

Wenn die Ausgangsdatenangabe aus dem Subtrahierer 565, das heißt die Differenzdatenangabe, ebenso groß wie die im Register 551 gespeicherte Wortzahl ist, welche den Voll-Zustand anzeigt, dann setzt die Vergleichseinrichtung 550 ein Voll-Flag FF1. Wenn indessen die Ausgangsdatenangabe aus dem Subtrahierer 565 ebenso groß wie die im Register 552 gehaltene Wortzahl ist, welche den Leer-Zustand anzeigt, dann setzt die Vergleichseinrichtung 550 ein Leer-Flag EF1. In gleichartiger Weise wird im in Fig. 39 dargestellten Ausgangsspeicher für eine Codefolge variabler Länge 484 die gleiche Operation wie im Eingangsspeicher 483 ausgeführt und das Voll-Flag FF2 oder das Leer-Flag EF2 aus der Vergleichseinrichtung 558 ausgegeben.

Die Steuereinrichtung 545 steuert die Datenübertragung zwischen den FIFO-Schaltungen 542 und 543, dem Ausgangsspeicher für eine Codefolge variabler Länge 484 und dem Eingangsspeicher für eine Codefolge variabler Länge 483 auf Grundlage sowohl des Flag-Ausgangssignals 42b aus der in Fig. 44 dargestellten FIFO-Schaltung 542, des Flag-Ausgangssignals 43b aus der FIFO-Schaltung 543 als auch auf Grundlage der Flags FF1, EF1, FF2 und EF2 aus den Vergleichseinrichtungen 550 und 558. Die Steuerung der Datenübertragung wird in der folgenden Weise ausgeführt.

Wenn das Codieren mit variabler Länge ausgeführt wird, dann werden Daten zwischen der FIFO-Schaltung 542 und dem Ausgangsspeicher für eine Codefolge variabler Länge 484 übertragen. Zu dieser Zeit wird in der Schaltung zum Erzeugen/Trennen einer Codefolge variabler Länge 505 die Datenübertragung gesteuert, um die FIFO-Schaltung 542 nicht in den Nicht-Überlaufzustand zu versetzen, das heißt, eine Leseoperation ohne die Speicherung von Daten zu verhindern. Ferner werden Daten übertragen, um keinen Überlauf des Ausgangsspei-

chers für eine Codefolge variabler Länge 484 zu verursachen, das heißt, um eine Schreiboperation in den Ausgangsspeicher für eine Codefolge variabler Länge 484 mit bereits gespeicherten gültigen Daten zu verhindern.

- 5 Beim Decodieren mit variabler Länge werden Daten zwischen der FIFO-Schaltung 543 und dem Eingangsspeicher für eine Codefolge variabler Länge 483 übertragen. Zu dieser Zeit wird die Datenübertragung gesteuert, um einen Überlauf der FIFO-Schaltung 543 und einen Nicht-Überlauf des Ausgangsspeichers für eine Codefolge variabler Länge 484 zu verhindern.

## 5. Bildpufferspeicher

10

### (1) Systemaufbau (Fig. 53 und 54)

Fig. 53 zeigt einen Grundaufbau von einer Speicherzellenanordnung im Bildpufferspeicher. Der in Fig. 1 dargestellte Bildpufferspeicher 51 weist beispielsweise den Aufbau der Fig. 53 auf.

- 15 Unter Bezugnahme auf Fig. 53 umfaßt der Bildpufferspeicher 54 (oder 55) insgesamt 32 Ebenen von Speicherzellenanordnungen 701 bis 732 (oder 801 bis 832). Wenn eine Zeilenadresse RA und eine Spaltenadresse CA angelegt werden, dann wird 1 Bit der Daten aus jeder der Speicherzellenanordnungen 701 bis 732 gelesen (oder dort eingeschrieben). Wenn beispielsweise eine Zeilenadresse RA1 und eine Spaltenadresse CA1 angelegt werden, dann werden insgesamt 32 Bit der Daten aus den Speicherzellenanordnungen 701 bis 732 gelesen.

- 20 Im allgemeinen werden 8 Bit der Daten benötigt, um ein Pixel darzustellen. Daher können durch insgesamt 32 Bit der Daten vier Pixel PC1 bis PC4 dargestellt werden (vergleiche Fig. 53). Mit anderen Worten, durch Anlegen von einer Zeilenadresse RA und einer Spaltenadresse CA können die Daten für vier Pixel PC1 bis PC4 verarbeitet werden.

- Fig. 54 stellt ein Blockschaltbild des Bildpufferspeichers dar. Unter Bezugnahme auf Fig. 54 umfaßt der Bildpufferspeicher 54 (oder 55) eine Steuerschaltung 741 und Speicherzellenanordnungen 701 bis 732 (oder 801 bis 832). Im Falle von SDRAMs ist jede der Speicherzellenanordnungen 701 bis 732 in zwei Bänke BK1 und BK1 eingeteilt. Bezüglich jeder der Speicherzellenanordnungen 701 bis 732 sind ein Abtastverstärker 742 und ein Eingangs-/Ausgangspuffer 743 vorgesehen. Daher werden 32 Bit von Daten PD1 bis PD32 in die und aus den Speicherzellenanordnungen 701 bis 732 geschrieben und gelesen.

- 30 Die Steuerschaltung 741 empfängt ein Adreßsignal ADR sowie ein Steuersignal Sc und erzeugt ein Steuersignal zum Zugreifen auf die Speicherzellenanordnungen 701 bis 732.

### (2) Ein DRAMs verwendendes Beispiel (erstes Beispiel: Fig. 55)

- 35 Fig. 55 zeigt einen Systemaufbau eines Bildpufferspeichers 54, welcher DRAMs verwendet. Unter Bezugnahme auf Fig. 55 umfaßt der Bildpufferspeicher 54 DRAM-Speicherzellenanordnungen 701 bis 732, welche insgesamt 32 Ebenen bilden. Der Bildpufferspeicher 54 empfängt ein Adreßsignal ADR aus einer DRAM-Steuereinrichtung 740 durch einen Adreßbus AB. Der Bildpufferspeicher 54 ist mit einem Pixeldatenbus PD verbunden, und die Daten werden durch den Pixeldatenbus PB eingegeben/ausgegeben.

- 40 Eine DRAM-Steuereinrichtung 742 ist beispielsweise in einer Steuereinheit der in Fig. 1 dargestellten Vorrichtung zur Bildkompression/-expansion vorgesehen. Insbesondere ist eine DRAM-Steuereinrichtung 743 in dem in Fig. 9 dargestellten Prozessor zur Gesamtsteuerung 11 vorgesehen.

- Fig. 56 stellt eine Pixelanordnung dar, welche die Beziehung zwischen den Daten, die im in Fig. 55 gezeigten Bildpufferspeicher 54 gespeichert sind, und den Pixeln auf dem Bildschirm angibt. Wie vorstehend beschrieben, kann durch Anlegen einer Zeilenadresse RA und einer Spaltenadresse CA an den Bildpufferspeicher 54 auf 32 Bit der vier Pixel angehenden Daten zugegriffen werden. Bei dem in Fig. 55 dargestellten Systemaufbau können durch eine Zeilenadresse und eine Spaltenadresse vier Pixel (zum Beispiel PC0—PC3) in der horizontalen Richtung auf einem Bildschirm SCN angegeben werden, wie in Fig. 56 gezeigt.

- 50 Fig. 58 stellt eine Pixelanordnung zur Beschreibung des Adreßsignals durch die in Fig. 55 dargestellte DRAM-Steuereinrichtung 740 dar. Unter Bezugnahme auf Fig. 58 werden durch die Spaltenadresse des Bildpufferspeichers 54 Pixel in der horizontalen Richtung des Bildschirms SCN festgelegt. Indessen werden durch das Zeilenadreßsignal Pixel in der vertikalen Richtung des Bildschirms SCN festgelegt. In Fig. 58 stellt das Bezugszeichen O ein Pixel dar.

- Der in Fig. 55 dargestellte Prozessor zur Gesamtsteuerung 11 erzeugt ein Zeilenadreßsignal RA und ein Spaltenadreßsignal CA gemäß einem darin gespeicherten Unterprogramm. Das Zeilen- und das Spaltenadreßsignal RA und CA werden als Adreßsignal ADR an den Adreßbus AB gelegt. Das Adreßsignal ADR wird durch den Adreßbus AB an den Bildpufferspeicher 54 gelegt. Zur gleichen Zeit sieht die DRAM-Steuereinrichtung 740 ein Steuersignal zum Steuern des Bildpufferspeichers 54 vor. Der Prozessor zur Gesamtsteuerung 11 überträgt beim Datenschieben zu speichernde Daten durch den Pixeldatenbus PD zum Bildpufferspeicher 54. Der 60 Prozessor zur Gesamtsteuerung 11 empfängt bei der Datenleseoperation die im Bildpufferspeicher 54 gespeicherten Daten durch den Pixeldatenbus PD.

- Fig. 59 stellt ein Zeitablaufdiagramm dar, welches die Leseoperation aus dem Bildpufferspeicher 54 unter der Steuerung der in Fig. 55 gezeigten DRAM-Steuereinrichtung 740 zeigt. Unter Bezugnahme auf Fig. 59 nimmt das Spaltenadreß-Strobesignal /RAS ab, nachdem das Zeilenadreßsignal RA1 angelegt wurde. Nach dem Anlegen des ersten Spaltenadreßsignals CA1 nimmt das Spaltenadreß-Strobesignal /CAS ab. Bei der Leseoperation wird ein Schreib-Freigabesignal /WE vom Hochpegel angelegt. Daher werden aus der durch die Adreßsignale RA1 und CA1 festgelegten Speicherzelle (nicht dargestellt) die gespeicherten Daten PD1 gelesen. Nach dem Lesen der Daten PD1 nimmt das Signal /CAS zu.



Nach dem Anlegen des nächsten Spaltenadreßsignals CA2 nimmt das Signal /CAS erneut ab. Folglich werden die Daten PD2 aus der durch die Adreßsignale RA1 und CA2 festgelegten Speicherzelle gelesen. Durch Wiederholen der gleichartigen Operationen werden die durch die Spaltenadreßsignale CA1, CA2, ... festgelegten Daten PD1, PD2, ... nacheinander aus einer durch das Zeilenadreßsignal RA1 festgelegten Zeile ausgegeben.

Unter Bezugnahme auf Fig. 58 wird ein Makroblock 750 durch insgesamt 64 ( $= 8 \times 8$ ) Pixel gebildet, das heißt: 8 in der horizontalen Richtung und 8 in der vertikalen Richtung auf dem Bildschirm SCN. Daher können die Pixeldaten im Makroblock 750 durch die Zeilenadreßsignale RA1 bis RA8 und die Spaltenadreßsignale CA4 und CA5 festgelegt werden.

Folglich erzeugt der in Fig. 5 dargestellte Prozessor zur Gesamtsteuerung 11 die Zeilen- und Spaltenadreßsignale gemäß einem gespeicherten Programm in der folgenden Art und Weise.

Zuerst werden nach dem Ausgeben des Zeilenadreßsignals RA1 die Spaltenadreßsignale CA4 und CA5 ausgegeben. Folglich kann auf die eine erste Zeile der Pixeldaten im Makroblock 750 zugegriffen werden (beispielsweise durch Lesen). Nach dem Anlegen des Zeilenadreßsignals RA2 werden dann die Spaltenadreßsignale CA4 und CA5 nacheinander angelegt. Folglich kann auf die zweite Zeile der Daten im Makroblock 750 zugegriffen werden. Eine gleichartige Operation wird für die achte Zeile des Makroblocks 750 wiederholt.

Um im Bildpufferspeicher 54 auf alle Pixel im Makroblock 750 zuzugreifen, sind folglich insgesamt 8 Anlegungen der Zeilenadressen RA1 bis RA8 und insgesamt 16 Anlegungen der Spaltenadressen CA erforderlich. In einem besonderen Beispiel erfordert die Anlegung einer Zeilenadresse RA 90 ns und zwei Anlegungen der Spaltenadressen CA und die Datenübertragung erfordern 120 ns. In diesem Fall erfordert die Festlegung von einem Makroblock 1680 ns ( $= (90 + 120) \times 8$ ).

Wenn indessen in der Einheit zur Bewegungsvoraussage ein Bewegungsvektor zu ermitteln ist, dann ist ein beispielsweise in Fig. 58 dargestellter Makroblock 751 notwendig. Insbesondere die im Makroblock 751 enthaltenen Pixeldaten werden durch die Zeilenadressen RA11 bis RA18 und die Spaltenadressen CA1 bis CA3 festgelegt. Die in Gebieten 752 und 753 gespeicherten Pixeldaten sind nicht notwendig, und sie werden verworfen, nachdem auf sie zum Zugreifen auf die Daten bezüglich der Pixel im Makroblock 751 zugegriffen wurde.

Um auf die Pixeldaten im Makroblock 751 zuzugreifen, werden daher insgesamt 8 Anlegungen der Zeilenadressen RA und insgesamt 24 Anlegungen der Spaltenadressen CA benötigt. Im Ergebnis werden für den Zugriff insgesamt 2160 ns benötigt.

### (3) Ein SDRAMs verwendendes Beispiel (zweites, drittes und viertes Beispiel)

Im folgenden werden drei Beispiele (zweites, drittes und viertes Beispiel) beschrieben werden, welche synchrone DRAMs (nachstehend als "SDRAMs" bezeichnet) als Bildpufferspeicher verwenden.

Fig. 60 stellt einen Systemaufbau des SDRAMs verwendenden Bildpufferspeichers 55 dar. Unter Bezugnahme auf Fig. 60 wird der Bildpufferspeicher 55 aus SDRAMs gebildet, welche Speicherzellenanordnungen 801 bis 832 umfassen, die in insgesamt 32 Ebenen unterteilt sind. Ein Prozessor zur Gesamtsteuerung 11 umfaßt einen Adreßgenerator 804 zum Erzeugen eines Adreßsignals ADR zum Zugriff auf den Bildpufferspeicher 55. Der Adreßgenerator 804 legt durch den Adreßbus AB ein Zeilenadreßsignal RA und ein Spaltenadreßsignal CA als Adreßsignal ADR an den Bildpufferspeicher 55. Die Daten, auf die zugegriffen wird, werden zum Bildpufferspeicher 55 durch den Pixeldatenbus PB übertragen. Der Prozessor zur Gesamtsteuerung sieht ein Systemuhr-Signal  $\Phi$ SC zum Erzeugen des Adreßsignals ADR zum Adreßgenerator 840 vor.

Beim Betrieb aktiviert der Prozessor zur Gesamtsteuerung 11 den Adreßgenerator 840 gemäß einem gespeicherten Programm. In Reaktion auf das Systemuhr-Signal  $\Phi$ SC sieht der Adreßgenerator 840 ein Adreßsignal ADR zum Zugreifen auf den SDRAM im Bildpufferspeicher 55 vor. Der Adreßgenerator 840 erzeugt ein Zeilenadreßsignal RA und ein Spaltenadreßsignal CA auf drei unterschiedliche Arten, welche nachstehend beschrieben werden (zweites, drittes und viertes Beispiel).

Fig. 61 stellt ein Zeitablaufdiagramm dar, welches einen Grundbetrieb (in diesem Beispiel den Lesebetrieb) des SDRAMs zeigt. Unter Bezugnahme auf Fig. 61 wird der SDRAM in Reaktion auf das aus dem Prozessor zur Gesamtsteuerung 11 angelegte Systemuhr-Signal  $\Phi$ SC betrieben. In Reaktion auf eine Abnahme des Signals /RAS wird das Zeilenadreßsignal RA1 erhalten, und danach wird in Reaktion auf eine Abnahme des Signals /CAS das Spaltenadreßsignal CA1 erhalten. In Reaktion auf das Systemuhr-Signal  $\Phi$ SC stellt der Adreßgenerator 840, mit dem Spaltenadreßsignal CA1 beginnend, nacheinander die Spaltenadreßsignale CA1 bis CA8 bereit. Diese Adreßsignale werden durch den Adreßbus AB an den Bildpufferspeicher 55 gelegt.

Daher werden im Datenlesebetrieb die gespeicherten Daten PD1 bis PD8 aus der Speicherzellenanordnung im SDFAM gelesen. Insbesondere werden in diesem in Fig. 61 dargestellten Beispiel 8 Spaltenadreßsignale erzeugt, wobei mit dem Spaltenadreßsignal CA1 begonnen wird, so daß von der durch das Zeilenadreßsignal RA1 festgelegten Zeile 8 Stücke der Daten PD1 bis PD8 gelesen werden.

#### (i) Zweites Beispiel (Fig. 62 und 63)

Fig. 62 zeigt eine Pixelanordnung zur Beschreibung des Adressierens durch den in Fig. 60 dargestellten Adreßgenerator 840 (zweites Beispiel), wie in Fig. 62 dargestellt, wobei die Pixel in der horizontalen Richtung auf dem Bildschirm SCN durch das Spaltenadreßsignal CA festgelegt werden, wohingegen die Pixel in der vertikalen Richtung durch das Zeilenadreßsignal RA festgelegt werden.

In diesem Beispiel werden ferner durch ein Zeilenadreßsignal RA und ein Spaltenadreßsignal CA insgesamt 32 Bit der Daten aus dem in Fig. 60 dargestellten Bildpufferspeicher 55 gelesen. Daher können vier Pixel durch 32 Bit der Daten dargestellt werden. In dem in Fig. 62 dargestellten Beispiel können ferner vier Pixel in der vertikalen Richtung (beispielsweise PC0—PC3) durch 32 Bit der Daten dargestellt werden, wie in Fig. 57 gezeigt.

Der Makroblock 860 umfaßt beispielsweise insgesamt 64 Pixel, das heißt 8 Pixel in der horizontalen Richtung und 8 Pixel in der vertikalen Richtung. Das Adressieren der Pixeldaten der Makrodaten 860 wird beim Lesen beispielsweise in der folgenden Weise ausgeführt.

Nach dem Anlegen einer Zeilenadresse RA4 wird eine Spaltenadresse CA1 angelegt. Der in Fig. 60 dargestellte Adreßgenerator 840 erzeugt in Reaktion auf das Systemuhr-Signal  $\Phi$ SC nacheinander, mit der Spaltenadresse CA1 beginnend, Spaltenadressen CA1 bis CA8. Daher wird von einer durch die Zeilenadresse RA4 festgelegten Zeile die erstere Hälfte der durch die Spaltenadressen CA1 bis CA8 festgelegten Daten nacheinander gelesen.

Nach dem Anlegen der Zeilenadresse RA5 wird dann die Spaltenadresse CA1 angelegt. In Reaktion auf das Systemuhr-Signal  $\Phi$ SC stellt der Adreßgenerator 840 nacheinander die Spaltenadreßsignale CA1 bis CA8 bereit. Daher kann die letztere Hälfte der Pixel im Makroblock 860 gelesen werden.

Fig. 63 stellt ein Zeitablaufdiagramm dar, welches den Betrieb des in Fig. 60 gezeigten Adreßgenerators zeigt. Um das Adressieren in der in Fig. 62 dargestellten Weise auszuführen, wird der in Fig. 60 dargestellte Adreßgenerator 840 wie in Fig. 63 dargestellt betrieben.

Unter Bezugnahme auf Fig. 63 wird in Reaktion auf eine Abnahme des Signals /RAS das Zeilenadreßsignal RA4 erhalten, und danach wird in Reaktion auf eine Abnahme des Signals /CAS das Spaltenadreßsignal CA1 erhalten. Der Adreßgenerator 840 (nicht dargestellt) stellt, beginnend mit dem erhaltenen Spaltenadreßsignal CA1, nacheinander die Spaltenadreßsignale CA1 bis CA8 bereit. Diese Adreßsignale RA4 und CA1 bis CA8 werden durch den in Fig. 60 dargestellten Adreßbus AB an den Bildpufferspeicher 55 gelegt. Als Ergebnis werden im Lesebetrieb die gespeicherten Daten PD1, PD2, ... nacheinander aus dem Bildpufferspeicher 55 ausgegeben.

Wie vorstehend beschrieben, werden aus jeder der Speicherzellenanordnungen 801 bis 832 1 Bit der Daten gelesen, da der Bildspeicher 55 insgesamt 32 Ebenen umfaßt. Ein in Fig. 63 dargestelltes Stück der Daten PD1 entspricht nämlich 32 Bit der Daten. Daher wird bemerkt, daß durch ein Stück der Daten PD1 vier Pixel in der vertikalen Richtung auf dem in Fig. 62 dargestellten Bildschirm SCN dargestellt werden können.

In Reaktion auf eine Abnahme des Signals /RAS wird dann das Zeilenadreßsignal RA5 erhalten, und in Reaktion auf eine Abnahme des Signals /CAS wird das Spaltenadreßsignal CA1 erhalten. Der Adreßgenerator 840 stellt, mit dem Spaltenadreßsignal CA1 beginnend, nacheinander die Adreßsignale CA1 bis CA8 (nicht dargestellt) bereit. Daher werden die Daten PD11 bis PD18 zum Darstellen der Pixel in der letzteren Hälfte des Makroblocs 860 nacheinander gelesen.

Auf diese Weise werden nur zwei Zeilenadreßsignale (das heißt RA4 und RA5) und nur zwei Spaltenadreßsignale (zweimal das Signal CA1) für die Pixel in einem Makroblock 860 benötigt, wobei die zum Adressieren benötigte Zeit im Vergleich zu dem in Fig. 58 dargestellten Beispiel verkleinert werden kann.

Wenn beispielsweise 25 ns zum Anlegen von einer Zeilenadresse sowie 200 ns zum Anlegen von einer Spaltenadresse CA und die Datenübertragung benötigt werden, dann erfordert eine Adressierung insgesamt 450 ns.

Ein anderes in Fig. 62 dargestelltes Pixelgebiet 862 stellt dasjenige Pixelgebiet dar, welches in der Einheit zur Bewegungsvoraussage bearbeitet wird. Um die Daten der Pixel im Gebiet 862 zu lesen, wird das Zeilenadreßsignal RA1 und dann das Spaltenadreßsignal CA11 angelegt. Daher werden aus der durch die Zeilenadresse RA1 festgelegten Zeile die Daten der Pixel im ersten Drittel des Pixelgebiets 862 nacheinander gelesen. Durch aufeinanderfolgendes Anlegen der Zeilenadreßsignale RA2 und RA3 kann das Lesen der verbliebenen Gebiete im Pixelgebiet 862 ausgeführt werden.

#### (ii) Drittes Beispiel (Fig. 64)

Fig. 64 stellt ein Zeitablaufdiagramm dar, welches ein anderes Beispiel des Betriebs des in Fig. 60 dargestellten Adreßgenerators 840 zeigt. Unter Bezugnahme auf Fig. 64 wird in diesem Beispiel das Spaltenadreßsignal CA (nicht dargestellt) vom Adreßgenerator 840 erzeugt, nachdem das Zeilenadreßsignal RA1 und das Spaltenadreßsignal CA1 in Reaktion auf das Systemuhr-Signal  $\Phi$ SC erhalten wurden. Zwei Spaltenadreßsignale CA1 und CA2 sind nämlich an den Bildpufferspeicher 55 zu legen. Folglich werden aus der durch die Zeilenadresse RA1 festgelegten Zeile 32 Bit der durch die Spaltenadresse CA1 festgelegten Daten PD1 und werden 32 Bit der durch die Spaltenadresse CA2 festgelegten Daten PD2 gelesen.

Da jede der Daten PD1 und PD2 32 Bit der Daten umfassen, können 8 Pixel in der horizontalen Richtung auf dem Bildschirm SCN dargestellt werden.

#### (iii) Viertes Beispiel (Fig. 65)

Fig. 65 stellt eine Pixelanordnung dar zur Beschreibung eines anderen Beispiels des Adressierens durch den in Fig. 60 gezeigten Adreßgenerator 840. Unter Bezugnahme auf Fig. 65 werden die Daten für die Pixel in der horizontalen Richtung auf dem Bildschirm SCN durch das Zeilenadreßsignal festgelegt, wohingegen die Daten der Pixel in der vertikalen Richtung durch das Spaltenadreßsignal festgelegt werden.

Um beispielsweise die Pixeldaten des Makroblocs 863 festzulegen, werden das Zeilenadreßsignal RA4 und dann das Spaltenadreßsignal CA1 angelegt. Der Adreßgenerator 840 erzeugt, beginnend mit dem Spaltenadreßsignal CA1, die Spaltenadressen CA1 bis CA8. Als Ergebnis werden die Daten der Pixel auf der linken Hälfte des Makroblocs 863 gelesen.

Nach dem Anlegen des Zeilenadreßsignals RA5 wird das Spaltenadreßsignal CA1 angelegt. Der Adreßgenerator 840 erzeugt die Spaltenadreßsignale CA1 bis CA8. Daher werden die Daten der Pixel auf der rechten Hälfte des Makroblocs 863 gelesen.

Bei der Verarbeitung in der Einheit zur Bewegungsvoraussage wird eine gleichartige Adressierung für die

Pixel im Pixelgebiet 865 bewirkt.

Auf diese Weise kann durch Anwenden des Bildpufferspeichers 55, welcher SDRAMs verwendet, wie im zweiten, dritten und vierten Beispiel dargestellt, die für die Adressierung notwendige Zeit verkleinert werden.

#### (4) Adreßgenerator (Fig. 66)

Fig. 66 stellt ein Blockschaltbild des in Fig. 60 gezeigten Adreßgenerators 840 dar. Unter Bezugnahme auf Fig. 66 umfaßt der Adreßgenerator 840: ein vertikales Anfangsadreßregister 841, ein Bildgrößenregister 842, ein Seitengrößenregister 843, ein horizontales Anfangsadreßregister 844, ein horizontales Größenregister 845, Register 846 und 847, einen horizontalen Zähler 848, eine Inkrementiereinrichtung 849, eine Ermittlungsschaltung für das rechte Ende 850, eine Ausgangs-Steuerschaltung 451, Auswahleinrichtungen 852 und 853, einen Addierer 854 und eine Auswahleinrichtung 855.

Es wird vorausgesetzt, daß die Register 841 und 845 Anfangsdaten entsprechend halten. Die im Register 841 gehaltene Anfangsadresse wird durch die Auswahleinrichtung 852, den Addierer 854 und die Auswahleinrichtung 855 als Zeilenadresse ausgegeben. Indessen wird die Anfangsadresse durch die Auswahleinrichtung 852 und den Addierer 854 in die Register 846 und 847 übertragen und darin gehalten.

Die im Register 844 gehaltene horizontale Ausgangsadresse wird durch die Inkrementiereinrichtung 849 zur Ausgangs-Steuerschaltung 851 für die Bitbreiten-Steuerung übertragen. Die Ausgangs-Steuerschaltung 851 stellt eine Spaltenadresse zur Verfügung, welche durch die Auswahleinrichtung 855 ausgegeben wird.

Die Inkrementiereinrichtung 859 inkrementiert die aus dem Register für die horizontale Ausgangsadresse 844 zugeführte horizontale Adresse aller 8 Zyklen des Systemuhr-Signals  $\Phi SC$ . Wenn die horizontale Adresse die Seitengröße überschreitet, dann inkrementiert die Inkrementiereinrichtung 849 die Zeilenadresse.

Die Daten des Registers 847 und die Datenangabe "1" werden dem Addierer 854 zugeführt, wobei das Additionsergebnis als Zeilenadresse vorgesehen wird und das Ergebnis ferner in das Register 847 überführt wird.

Wenn das rechte Ende der übertragenen Daten durch die Ermittlungsschaltung für das rechte Ende 850 ermittelt ist, dann werden die im Register 846 gehaltenen Daten und die Bildgrößendaten zum Addierer 854 durch die Auswahleinrichtungen 852 und 853 übertragen. Die das Additionsergebnis angegebenden Daten werden als Zeilenadresse durch die Auswahleinrichtung 855 ausgegeben und ferner in die Register 846 und 847 übertragen.

Fig. 67 stellt die Adressen-Seite-Entsprechung dar, welche die Beziehung zwischen den Zeilen- und Spaltenadressen, die vom in Fig. 66 dargestellten Adreßgenerator 840 vorgesehen werden, und den Seiten zeigt. Fig. 68 stellt indessen die Seiten-Pixel-Entsprechung dar, welche die Beziehung zwischen der in Fig. 67 gezeigten Seite und den Pixeln auf dem Bildschirm zeigt. Der in Fig. 60 dargestellte Bildpufferspeicher 55 speichert die Daten in der in Fig. 67 dargestellten Weise. Der Adreßgenerator 840 weist die im Bildpufferspeicher 55 gespeicherten Daten in der in Fig. 68 dargestellten Weise zu. Der in Fig. 66 dargestellte Adreßgenerator erzeugt das Zeilenadreßsignal RA und das Spaltenadreßsignal CA, derart daß das Adressieren in der in den Fig. 67 und 68 gezeigten Weise ausgeführt werden kann.

#### (5) Verfahren zum Abbilden

Im folgenden wird ein Verfahren zum Abbilden von jeder Pixel-Datenangabe des Bildschirms auf den Bildpufferspeicher beschrieben werden. Die nachstehend beschriebenen Verfahren können beim in Fig. 1 dargestellten Bildpufferspeicher 51 verwendet werden.

Zuerst wird ein Abbildungsverfahren beschrieben werden, welches eine Verkleinerung der Anzahl von Datenübertragungen erlaubt. Fig. 72 zeigt eine Darstellung eines ersten Verfahrens zum Abbilden von Pixeldaten des Bildschirms auf den Bildpufferspeicher. Unter Bezugnahme auf Fig. 72 wird das Datengebiet 861 mit zwei Pixeln in Längs- und Breitenrichtung als eine Einheit behandelt und eine Adresse des Bildpufferspeichers zugewiesen. Insbesondere werden entsprechend einer durch das Zeilenadreßsignal RA und das Spaltenadreßsignal CA spezifizierten Adresse Daten der vier im Datengebiet 861 enthaltenen Pixel im Bildpuffer gespeichert oder werden die Daten durch Festlegen der Adresse aus dem Bildpufferspeicher gelesen und die Daten zu und aus jedem der Blöcke durch den Pixeldatenbus übertragen. Wenn beispielsweise unter Bezugnahme auf Fig. 72 ein Datengebiet 862, das 8 Pixel in der horizontalen Richtung und 8 Pixel in der vertikalen Richtung umfaßt, zur Verwendung bei der Ermittlung des Bewegungsvektors übertragen wird, dann können unter Verwendung des Datengebiets 861 als Übertragungseinheit alle Pixeldaten im Datengebiet 862 durch 5 (horizontale)  $\times$  5 (vertikale) = 25 Übertragungsoperationen übertragen werden. Im Vergleich zu dem Beispiel der Fig. 58, bei welchem ein aus vier Pixeln in der horizontalen Richtung gebildetes Datengebiet als Übertragungseinheit verwendet wird, kann die Anzahl der Übertragungen um zwei verkleinert werden, und das Datengebiet 863, welches keine Übertragung verlangt, kann verkleinert werden. Daher kann durch Verwenden des vorstehend beschriebenen Abbildungsverfahrens die Anzahl der Übertragungen verkleinert werden, und die Bilddaten können mit hoher Geschwindigkeit übertragen werden.

Eine Übertragung eines aus 8 Pixeln in der horizontalen Richtung und 8 Pixeln in der vertikalen Richtung gebildeten Datengebiets unter Verwendung des vorstehend beschriebenen Abbildungsverfahrens wird beschrieben werden. Die Fig. 73 bis 76 stellen erste bis vierte Darstellungen der Übertragung des Datengebiets dar, welches aus horizontalen 8 Pixeln und vertikalen 8 Pixeln gebildet wird. Wenn ein Datengebiet 864 zu übertragen ist, welches aus 8 Pixeln in der horizontalen Richtung und 8 Pixeln in der vertikalen Richtung gebildet wird, dann gibt es in Abhängigkeit von der Position des Datengebiets 864 vier unterschiedliche Arten, die in den Fig. 73 bis 76 dargestellt sind.

In dem in Fig. 73 dargestellten Beispiel beträgt die notwendige Anzahl von Übertragungen:  $4 \text{ (horizontale)} \times 4 \text{ (vertikale)} = 16$ , wenn jede Pixeldatenangabe im Datengebiet 864 unter Verwendung eines zwei Pixel in horizontaler und vertikaler Richtung umfassenden Datengebiets als Übertragungseinheit zu übertragen ist. Im Beispiel der Fig. 74 und 75 beträgt die notwendige Anzahl der Übertragungen 20. Im Beispiel der Fig. 76 sind 25 Übertragungen erforderlich. Daher beträgt die zum Übertragen des Datengebiets 864 notwendige Durchschnittszahl von Übertragungen 20, 25. Im Vergleich zu der in Fig. 58 dargestellten Übertragung unter Verwendung von vier Pixeln in der horizontalen Richtung als Einheit kann die Anzahl der Übertragungen um etwa zwei verkleinert werden.

Ein Abbildungsverfahren wird beschrieben werden, bei welchem ein Datengebiet aus vier Pixeln in der horizontalen Richtung und zwei Pixeln in der vertikalen Richtung als ein Datenstück gebildet wird und ihm 1 Adresse zugewiesen ist. Fig. 77 zeigt eine zweite Darstellung, welche ein Verfahren zum Abbilden der Bilddaten des Bildschirms auf den Bildpufferspeicher darstellt. Wie in Fig. 77 gezeigt, wird 1 Adresse des Bildpufferspeichers zum Datengebiet 865 übertragen, welches aus vier Pixeln in der horizontalen Richtung und zwei Pixeln in der vertikalen Richtung als ein Datenstück ausgebildet ist. Zu dieser Zeit beträgt die Anzahl der benötigten Datenübertragungen:  $3 \text{ (horizontale)} \times 5 \text{ (vertikale)} = 15$ , wenn ein 9 Pixel in der horizontalen Richtung und 9 Pixel in der vertikalen Richtung umfassendes Datengebiet 866 unter Verwendung des Datengebiets 865 als Übertragungseinheit zu übertragen ist. Im Vergleich zu dem in Fig. 58 dargestellten Beispiel, bei welchem ein 8 Pixel in der horizontalen Richtung umfassendes Datengebiet als Einheit zur Übertragung verwendet wird, kann die Anzahl der Übertragungen um drei verkleinert werden, und das Gebiet 867, welches keine Datenübertragung verlangt, kann verkleinert werden.

Ein erstes Verfahren zum Abbilden eines ersten und eines zweiten Farbdifferenzbildes auf den Bildpufferspeicher wird beschrieben werden. Fig. 78 zeigt eine Darstellung des ersten Verfahrens zum Abbilden der ersten und der zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher. Dabei bedeuten die ersten Farbdifferenz-Pixeldaten die Farbdifferenz-Pixeldaten Cb, welche aus dem RGB-Format in das YUV-Format umgewandelt worden sind, und die zweiten Farbdifferenz-Pixeldaten bedeuten die Farbdifferenz-Pixeldaten Cr, die gleichartig umgewandelt worden sind. In Fig. 78 wird das aus vier Pixeln in der horizontalen Richtung gebildete Datengebiet 868 als ein Datenstück betrachtet und ihm 1 Adresse des Bildpufferspeichers zugewiesen. Erste Farbdifferenz-Pixeldaten 873 und zweite Farbdifferenz-Pixeldaten 869 werden zu vier Pixeln im Anschluß an vier Pixel in der horizontalen Richtung abwechselnd abgebildet. Die ersten Farbdifferenz-Pixeldaten 873 und die zweiten Farbdifferenz-Pixeldaten 869 sind miteinander korreliert. Bei der Ermittlung des Bewegungsvektors müssen beispielsweise die ersten und die zweiten Farbdifferenz-Pixeldaten 873 und 869 gleichzeitig verarbeitet werden, so daß die Datenübertragung gleichzeitig durchgeführt wird. Daher müssen alle Daten im Datengebiet 871 übertragen werden, wenn das erste Farbdifferenz-Pixeldaten-Gebiet 872 und das zweite Farbdifferenz-Pixeldaten-Gebiet 870 zu übertragen sind. Da das erste Farbdifferenz-Pixeldaten-Gebiet 872 und das zweite Farbdifferenz-Pixeldaten-Gebiet 870 abwechselnd abgebildet wird, können in diesem Fall die Farbdifferenz-Pixeldaten des Datengebiets 871 durch eine Adressenerzeugung entsprechend übertragen werden, so daß die für die Berechnung zur Adressenerzeugung benötigte Zeit verkleinert werden kann und die Bilddaten mit hoher Geschwindigkeit verarbeitet werden können.

Ein zweites Verfahren zum Abbilden der ersten und der zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher wird beschrieben werden. Fig. 79 zeigt eine Darstellung des zweiten Verfahrens zum Abbilden der ersten und der zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher. In Fig. 79 wird ein aus zwei Pixeln in der horizontalen Richtung und zwei Pixeln in der vertikalen Richtung gebildetes Datengebiet 874 als ein Datenstück betrachtet und zugewiesen. Erste Farbdifferenz-Pixeldaten 875 und zweite Farbdifferenz-Pixeldaten 879 werden zu zwei Pixeln im Anschluß an zwei Pixel in der horizontalen Richtung abwechselnd abgebildet. In diesem Beispiel werden alle Daten im Datengebiet 877 unter Verwendung des Datengebiets 874 als Übertragungseinheit übertragen, wenn das erste Farbdifferenz-Pixeldaten-Gebiet 878 und das zweite Farbdifferenz-Pixeldaten-Gebiet 876 zu übertragen sind. Daher können die Daten durch die Erzeugung von einer Adresse und durch eine kleinere Anzahl von Datenübertragungen übertragen werden.

Ein drittes Verfahren zum Abbilden der ersten und der zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher wird beschrieben werden.

Fig. 80 zeigt eine Darstellung des dritten Verfahrens zum Abbilden der ersten und der zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher. Im Beispiel der Fig. 80 wird ein aus vier Pixeln in der horizontalen Richtung gebildetes Datengebiet als ein Datenstück betrachtet, und demselben wird eine Adresse zugewiesen. Die ersten Farbdifferenz-Pixeldaten und die zweiten Farbdifferenz-Pixeldaten sind zu zwei Pixeln im Anschluß an zwei Pixel in der horizontalen Richtung abwechselnd angeordnet. Daher sind in dem mit einer Adresse versehenen Datengebiet der vier Pixel in der horizontalen Richtung zwei Pixel der ersten Farbdifferenz-Pixeldaten 880 und zwei Pixel der zweiten Farbdifferenz-Pixeldaten 881 enthalten. Wenn das erste Farbdifferenz-Pixeldaten-Gebiet 884 und das zweite Farbdifferenz-Pixeldaten-Gebiet 882 zu übertragen sind, dann ist es nur erforderlich, jede der Datenangaben im Datengebiet 883 durch Erzeugung von einer Adresse zu übertragen. Zu dieser Zeit würde die Anzahl der Übertragungen betragen:  $5 \text{ (horizontale)} \times 9 \text{ (vertikale)} = 45$ . Wenn die ersten Farbdifferenz-Pixeldaten und die zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher separat abgebildet werden und durch Verwenden des vier Pixel in der horizontalen Richtung umfassenden Datengebiets als Übertragungseinheit in der gleichartigen vorstehend beschriebenen Weise übertragen werden, dann beträgt die Anzahl der notwendigen Datenübertragungen für die beispielsweise in Fig. 80 gezeigten ersten und zweiten Farbdifferenz-Pixeldaten 54, und daher kann durch das dritte Abbildungsverfahren die Anzahl der Übertragungen um neun verkleinert werden.

Ein viertes Verfahren zum Abbilden der ersten und zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher wird beschrieben werden. Fig. 81 zeigt eine Darstellung des vierten Verfahrens zum Abbilden der ersten

und der zweiten Farbdifferenz-Pixeldaten auf dem Bildpufferspeicher. Wie in Fig. 81 gezeigt, wird ein aus vier Pixeln in der horizontalen Richtung und aus zwei Pixeln in der vertikalen Richtung gebildetes Datengebiet als ein Datenstück betrachtet, und ihm wird eine Adresse zur Abbildung auf den Bildpufferspeicher zugewiesen. Die ersten Farbdifferenz-Pixeldaten und die zweiten Farbdifferenz-Pixeldaten sind zu zwei Pixeln im Anschluß an zwei Pixel in der horizontalen Richtung abwechselnd angeordnet. Daher umfassen in dem als ein Datenstück abgebildeten Datengebiet die ersten Farbdifferenz-Pixeldaten 885 zwei Pixel in der horizontalen Richtung und zwei Pixel in der vertikalen Richtung, ebenso wie die zweiten Farbdifferenz-Pixeldaten 886 zwei Pixel in der horizontalen Richtung und zwei Pixel in der vertikalen Richtung enthalten. Wenn die ersten Farbdifferenz-Pixeldaten 889 und die zweiten Farbdifferenz-Pixeldaten 887 zu übertragen sind, dann ist es erforderlich, jede der Datenangaben im Datengebiet 888 unter Verwendung des 4 Pixel in der horizontalen Richtung und zwei Pixel in der vertikalen Richtung umfassenden Datengebiets als Übertragungseinheit zu übertragen. Zu dieser Zeit können die Daten durch die Erzeugung von einer Adresse übertragen werden, da die ersten Farbdifferenz-Pixeldaten und die zweiten Farbdifferenz-Pixeldaten abwechselnd angeordnet sind, und die Anzahl der Übertragungen kann verkleinert werden, da ein rechteckiges Datengebiet als Übertragungseinheit verwendet wird.

Die Wirkungen des ersten bis vierten Verfahrens zum vorstehend beschriebenen Abbilden der ersten und zweiten Farbdifferenz-Pixeldaten auf den Bildpufferspeicher werden erläutert. Ein Betrieb des in Fig. 9 dargestellten Prozessors zur Gesamtsteuerung beim Codieren der Zwei-Weg-Voraussage wird beispielhaft beschrieben werden. Fig. 82 zeigt den Betrieb, der im Prozessor zur Gesamtsteuerung beim Codieren der Zwei-Weg-Voraussage ausgeführt wird.

Unter Bezugnahme auf Fig. 82 wird in einem Schritt 891 in einem Eingangsbild-Schreibprozeß eine erste Adresse erzeugt. Die Erzeugung der Adresse wird für Helligkeits-Pixeldaten Y, erste Farbdifferenz-Pixeldaten Cb und zweite Farbdifferenz-Pixeldaten Cr ausgeführt. Diese Daten werden aus der Steuereinheit 2 in den Bildpufferspeicher 51 in der in Fig. 1 dargestellten Vorrichtung zur Bilddatenverarbeitung übertragen.

In einem Schritt 892 wird beim Lesen zur Bewegungsvoraussage die Erzeugung der zweiten Adresse ausgeführt. Dieser Prozeß wird für die Helligkeits-Pixeldaten Y ausgeführt, und die Daten werden aus dem Bildpufferspeicher 51 zur Steuereinheit 2 übertragen.

In einem Schritt 893 wird dann beim Lesen des Bereichs der Bewegungssuche die Erzeugung der dritten Adresse ausgeführt. Dabei wird der Prozeß für die Helligkeits-Pixeldaten Y ausgeführt, und die Daten werden aus dem Bildpufferspeicher 51 zur Einheit zur Bewegungsvoraussage 41 übertragen.

In einem Schritt 984 wird dann der Codiermodus festgelegt. Danach wird beim Lesen des Bildes für die Vorwärtsvoraussage in einem Schritt 895 die Erzeugung der vierten Adresse ausgeführt. Dabei wird der Prozeß für drei unterschiedliche Pixeldaten Y, Cb und Cr ausgeführt, und die Daten werden aus dem Bildpufferspeicher 51 zur Pixelverarbeitungseinheit 3 übertragen.

In einem Schritt 869 wird beim Lesen des Bildes für die Rückwärtsvoraussage die Erzeugung der fünften Adresse ausgeführt. Dabei wird der Prozeß an drei unterschiedlichen Pixeldaten Y, Cb und Cr ausgeführt, und die Daten werden aus dem Bildpufferspeicher 51 zur Pixelverarbeitungseinheit 3 übertragen.

In einem Schritt 893 wird beim Lesen des zu codierenden Bildes die Erzeugung der sechsten Adresse ausgeführt. Dabei wird der Prozeß an drei unterschiedlichen Pixeldaten Y, Cb und Cr ausgeführt, und die Daten werden aus dem Bildpufferspeicher 51 zur Pixelverarbeitungseinheit 3 übertragen.

Die nachstehende Tabelle zeigt das Ergebnis eines Vergleichs zwischen den Ausführungsformen, welche die in den Fig. 78 bis 81 dargestellten ersten bis vierten Abbildungsverfahren verwenden, und einem Vergleichsbeispiel hinsichtlich der Anzahl von vorstehend beschriebenen Operationen, die im Prozessor zur Gesamtsteuerung ausgeführt werden. Im Vergleichsbeispiel werden drei Typen von Pixeldaten Y, Cb und Cr auf den Bildpufferspeicher 51 separat abgebildet und Adressen für alle Pixeldaten erzeugt. Das Bezugszeichen C in nachstehender Tabelle 1 bezieht sich im allgemeinen auf die ersten und zweiten Farbdifferenz-Pixeldaten Cb und Cr.

Tabelle 1

Inhalt der Übertragung	Vergleichsbeispiel		Ausführungsform	
	Komponente	Anzahl von Operationen	Komponente	Anzahl von Operationen
Schreiben des Eingangsbildes	Y, Cb, Cr	3	Y, C	2
Lesen des Bildes zur Bewegungsvoraussage	Y	1	Y	1
Lesen des Bereichs der Bewegungssuche	Y	1	Y	1
Lesen des Bildes zur Vorwärtsvoraussage	Y, Cb, Cr	3	Y, C	2
Lesen des Bildes zur Rückwärtsvoraussage	Y, Cb, Cr	3	Y, C	2
Lesen des zu codierenden Bildes	Y, Cb, Cr	3	Y, C	2
Insgesamt		14		10

Wie aus der Tabelle zu ersehen ist, ist die Anzahl der Adressen-Erzeugungen bei den Ausführungsformen um vier verkleinert. Insbesondere beim Prozeß zur Erzeugung einer Adresse im Vergleichsbeispiel ist es notwendig, die Operationen an jedem der drei unterschiedlichen Typen von Pixeldaten Y, Cb und Cr auszuführen und die entsprechenden Daten in den Registern der in Fig. 9 dargestellten Steuereinheit zur Bilddatenübertragung 15 abzulegen. Indessen können bei der vorstehend beschriebenen Ausführungsform die Gesamtzahl von vier Operationen und die zum Belegen der Register benötigte Zeit verkleinert werden, da die Adressen für die ersten Farbdifferenz-Pixeldaten Cb und die zweiten Farbdifferenz-Pixeldaten Cr gemeinsam erzeugt werden. Im Prozessor zur Gesamtsteuerung müssen verschiedene Prozesse ausgeführt werden, wie beispielsweise die Steuerung der Codemenge, verschiedene Bestimmungen bezüglich des Codierens und andere anwenderspezifische Operationen. Wenn daher die Anzahl der Operationen durch diese verschiedenen Operationen zunimmt, dann hat die vorstehend beschriebene Reduzierung bei der Adreßberechnung eine wesentliche Auswirkung auf das Codieren auf Echtzeitbasis.

Obwohl die vorliegende Erfindung detailliert beschrieben und dargestellt worden ist, ist zu verstehen, daß dieselbe nur veranschaulichend und beispielhaft ist und keiner Beschränkung unterliegt, wobei der Inhalt und der Bereich der vorliegenden Erfindung nur durch die beigefügten Ansprüche beschränkt wird.

#### Patentansprüche

1. Vorrichtung zur Bilddatenverarbeitung, umfassend:

eine Steuereinrichtung (11) zum Steuern einer Pipelineverarbeitung von einer Mehrzahl von vorbestimmten Prozessen zur Bilddatenkompression und/oder Bilddatenexpansion und eine Bildverarbeitungseinrichtung (12) zum Ausführen eines vorbestimmten Prozesses der Mehrzahl von vorbestimmten Prozessen in Reaktion auf ein aus der Steuereinrichtung angelegtes Steuersignal.

2. Bilddatenvorrichtung nach Anspruch 1, bei welcher die Steuereinrichtung umfaßt:

eine Einrichtung zur Steuerung der Pipelineverarbeitung (11) zum Steuern der Pipelineverarbeitung einer diskreten Kosinusumwandlung, einer Quantisierung und einer Verarbeitung von Daten variabler Länge; und bei welcher

die Bildverarbeitungseinrichtung umfaßt:

eine Einrichtung zur Verarbeitung einer variablen Länge (12) zum Ausführen der Verarbeitung von Daten variabler Länge in Reaktion auf ein aus der Einrichtung zur Steuerung der Pipelineverarbeitung angelegtes Steuersignal.

3. Bilddatenvorrichtung nach Anspruch 2, ferner umfassend:

eine Einrichtung zur Pixeldatenspeicherung (51) zum Speichern von zu komprimierenden Pixeldaten; einen Pixeldatenbus (PB) zum Übertragen der in der Einrichtung zur Pixeldatenspeicherung gespeicherten Pixeldaten;

eine Pixelcode-Ausgabeeinrichtung (3) zum Ausführen der diskreten Kosinsumwandlung und der Quantisierung an den Pixeldaten, die durch den Pixeldatenbus übertragen werden, um einen Pixelcode auszugeben; und  
 einen Pixelcodebus (CB) zum Übertragen des aus dem Pixelcodeausgang ausgegebenen Pixelcodes; bei welcher  
 die Einrichtung zur Verarbeitung einer variablen Länge umfaßt:  
 eine Verarbeitungseinrichtung zum Codieren mit variabler Länge (12) zum Ausführen des Codierens mit variabler Länge an dem durch den Pixelcodebus übertragenen Pixelcode und Vorsehen eines Codes variabler Länge; und bei welcher  
 die Einrichtung zur Steuerung der Pipelineverarbeitung die Pipelineverarbeitung bezüglich des Pixeldatenbusses, der Pixelcode-Ausgabeeinrichtung, des Pixelcodebusses und der Verarbeitungseinrichtung zum Codieren mit variabler Länge steuert. 10

4. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 2, ferner umfassend:  
 eine Einrichtung zum Speichern des Codes variabler Länge (51) zum Speichern eines komprimierten Codes variabler Länge; und einen Bus des Codes variabler Länge (PB) zum Übertragen des in der Einrichtung zum Speichern des Codes variabler Länge gespeicherten Codes variabler Länge; bei welcher  
 die Einrichtung zur Verarbeitung einer variablen Länge umfaßt:  
 eine Verarbeitungseinrichtung zum Decodieren mit variabler Länge (12) zum Ausführen eines Decodierens mit variabler Länge an dem Code variabler Länge, welcher durch den Bus des Codes variabler Länge zum Ausgeben eines Pixelcodes übertragen wird; 15  
 wobei  
 die Vorrichtung zur Bilddatenverarbeitung ferner umfaßt:  
 einen Pixelcodebus (CB) zum Übertragen der Pixelcodeausgabe aus der Verarbeitungseinrichtung zum Decodieren mit variabler Länge; und  
 eine Pixeldaten-Ausgabeeinrichtung (3) zum Ausführen einer inversen Quantisierung und einer inversen diskreten Kosinsumwandlung an dem Pixelcode, der zum Ausgeben von expandierten Pixeldaten durch den Pixelcodebus übertragen wird; bei welcher die Einrichtung zur Steuerung der Pipelineverarbeitung die Pipelineverarbeitung bezüglich des Busses des Codes variabler Länge, der Verarbeitungseinrichtung zum Decodieren mit variabler Länge, des Pixelcodebusses und der Pixeldaten-Ausgabeeinrichtung steuert. 20

5. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 2, ferner umfassend:  
 eine Pixelverarbeitungseinrichtung (3) zum Ausführen der diskreten Kosinsumwandlung und der Quantisierung an den Pixeldaten in einem Datenkompressionsmodus und zum Ausführen der inversen Quantisierung und der inversen diskreten Kosinsumwandlung an einem Pixelcode in einem Datenexpansionsmodus, bei welcher  
 die Einrichtung zur Verarbeitung einer variablen Länge ferner umfaßt 25  
 eine Verarbeitungseinrichtung zum Codieren/Decodieren mit variabler Länge (12) zum Ausführen des Codierens mit variabler Länge am Pixelcode im Datenkompressionsmodus und zum Ausführen des Decodierens mit variabler Länge am Code variabler Länge im Datenexpansionsmodus; wobei  
 die Vorrichtung zur Bilddatenverarbeitung ferner umfaßt:  
 einen ersten Datenbus (PB) zum Übertragen von Bilddaten und eines Codes variabler Länge zwischen der Pixelverarbeitungseinrichtung und der Verarbeitungseinrichtung zum Codieren/Decodieren mit variabler Länge; und 30  
 einen zweiten Datenbus (CB) zum Übertragen des Pixelcodes zwischen der Pixelverarbeitungseinrichtung und der Verarbeitungseinrichtung zum Codieren/Decodieren mit variabler Länge; bei welcher  
 die Einrichtung zur Steuerung der Pipelineverarbeitung die Pipelineverarbeitung bezüglich der Pixelverarbeitungseinrichtung, der Verarbeitungseinrichtung zum Codieren/Decodieren mit variabler Länge und des ersten und des zweiten Datenbusses steuert. 35

6. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 5, ferner umfassend:  
 eine mit dem ersten Datenbus verbundene Speichereinrichtung (51) zum Speichern der zu komprimierenden Pixeldaten und eines komprimierten Codes variabler Länge; sowie 40  
 eine mit dem ersten Datenbus verbundene Speichereinrichtung zur Eingabe/Ausgabe der Bilddaten (81) zum Speichern extern eingegebener Bilddaten und extern auszugebender Bilddaten.

7. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 5, ferner umfassend:  
 eine mit dem ersten Datenbus verbundene erste Verarbeitungseinrichtung zur Bewegungsvoraussage (43) zum Ausführen einer Bewegungsvoraussage in Vorwärtsrichtung; und 45  
 eine mit dem ersten Datenbus verbundene zweite Verarbeitungseinrichtung zur Bewegungsvoraussage (44) zum Ausführen einer Bewegungsvoraussage in Rückwärtsrichtung.

8. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 7, ferner umfassend:  
 eine mit dem ersten Datenbus verbundene erste Bilddaten-Speichereinrichtung (52) zum Speichern von Bilddaten für die Bewegungsvoraussage in der Vorwärtsrichtung; und 50  
 eine mit dem ersten Datenbus verbundene zweite Bilddaten-Speichereinrichtung (53) zum Speichern von Bilddaten für die Bewegungsvoraussage in der Rückwärtsrichtung.

9. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 5, ferner umfassend:  
 eine mit dem ersten Datenbus verbundene Verarbeitungseinrichtung zur Bewegungsvoraussage (46) zum Ausführen der Bewegungsvoraussage; bei welcher 55  
 die Einrichtung zur Steuerung der Pipelineverarbeitung ferner eine direkt mit der Verarbeitungseinrichtung zur Bewegungsvoraussage verbundene Einrichtung zur Steuerung der Bewegungsvoraussage (13) zum Steuern des Betriebs der Verarbeitungseinrichtung zur Bewegungsvoraussage umfaßt. 60

65



10. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 2, ferner umfassend:

eine erste Pixeldaten-Speichereinrichtung (51) zum Speichern von zu komprimierenden Pixeldaten;  
einen ersten Pixeldatenbus (PB) zum Übertragen der in der ersten Pixeldaten-Speichereinrichtung gespeicherten Pixeldaten;

5 eine Bildverarbeitungseinrichtung (33), welche die durch den ersten Pixeldatenbus übertragenen Pixeldaten empfängt, um unter Verwendung der übertragenen Pixeldaten Referenzdaten zur Bewegungsvoraussage zu gewinnen, und welche ein erstes und ein zweites Datenport (P1, P2) zum Ausgeben der Referenzdaten zur Bewegungsvoraussage aufweist;  
einen zweiten Pixeldatenbus (LDB) zum Übertragen der Referenzdaten zur Bewegungsvoraussage, welche aus dem zweiten Datenport der Bildverarbeitungseinrichtung ausgegeben werden;  
10 eine zweite Pixeldaten-Speichereinrichtung (56) zum Speichern der durch den zweiten Bus übertragenen Referenzdaten zur Bewegungsvoraussage; und  
eine Verarbeitungseinrichtung zur Bewegungsvoraussage (47) zum Ausführen der Bewegungsvoraussage durch Verwenden der Referenzdaten zur Bewegungsvoraussage, welche in der zweiten Pixeldaten-Speichereinrichtung gespeichert sind und durch den zweiten Pixeldatenbus zum Erzeugen eines Bewegungsvektors übertragen werden; bei welcher  
15 die Einrichtung zur Steuerung der Pipelineverarbeitung eine Einrichtung (23) zum Steuern der Pipelineverarbeitung bezüglich des ersten und des zweiten Pixeldatenbusses, der Pixelverarbeitungseinrichtung und der Verarbeitungseinrichtung zur Bewegungsvoraussage umfaßt.

20 11. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 10, bei welcher die Einrichtung zur Steuerung der Pipelineverarbeitung ferner eine Systemsteuereinrichtung (114) umfaßt, welche ein Ausgangssteuersignal des ersten und des zweiten Datenports der Pixelverarbeitungseinrichtung ausgibt.

12. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 10, bei welcher  
25 die Pixelverarbeitungseinrichtung eine erste Pixelverarbeitungseinrichtung (34) umfaßt, welche die durch den ersten Pixeldatenbus übertragenen Pixeldaten zum Gewinnen von Referenzdaten zur Bewegungsvoraussage unter Verwendung der übertragenen Pixeldaten empfängt und welche ein drittes Datenport (P3) zum Ausgeben der Referenzdaten zur Bewegungsvoraussage sowie eine Mehrzahl von vierten Datenports (P4, P5) zum Verteilen und Ausgeben der Referenzdaten zur Bewegungsvoraussage aufweist;  
30 der zweite Pixeldatenbus eine Mehrzahl von dritten Pixeldatenbussen (LDB1, LDB2) umfaßt, welche den vierten Datenports entsprechend zum Übertragen der aus den vierten Datenports ausgegebenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind;  
die zweite Pixeldaten-Speichereinrichtung eine Mehrzahl von dritten Pixeldaten-Speichereinrichtungen (56a, 56b) umfaßt, welche den vierten Datenports entsprechend zum Speichern der durch die dritten  
35 Pixeldatenbusse übertragenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind;  
die Verarbeitungseinrichtung zur Bewegungsvoraussage eine Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage (47a, 47b) umfaßt, die den vierten Datenports entsprechend zum Ausführen der Bewegungsvoraussage unter Verwendung der Referenzdaten zur Bewegungsvoraussage vorgesehen sind, welche in den dritten Pixeldaten-Speichereinrichtungen gespeichert sind und durch die dritten  
40 Pixeldatenbusse übertragen werden; und bei welcher die Einrichtung zur Steuerung der Pipelineverarbeitung eine Einrichtung zum Steuern der Pipelineverarbeitung bezüglich des ersten Pixeldatenbusses, der ersten Pixelverarbeitungseinrichtung, der Mehrzahl von dritten Pixeldatenbussen und der Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage umfaßt.

45 13. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 12, welche ferner einen Adreßbus umfaßt, der mit der Mehrzahl der zweiten Pixeldaten-Speichereinrichtungen verbunden ist; bei welcher die erste Pixelverarbeitungseinrichtung eine zweite Pixelverarbeitungseinrichtung (34) zum Verteilen und zum Ausgeben der Referenzdaten zur Bewegungsvoraussage gemäß einem Typ der Referenzdaten zur Bewegungsvoraussage aus der Mehrzahl von vierten Datenports (P4, P5) umfaßt;  
50 die Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage eine Mehrzahl von zweiten Verarbeitungseinrichtungen zur Bewegungsvoraussage (47a, 47b) zum Erzeugen eines Bewegungsvektors gemäß dem Typ der Referenzdaten zur Bewegungsvoraussage umfaßt; und bei welcher die Einrichtung zur Steuerung der Pipelineverarbeitung eine Einrichtung (25) zum Steuern der Pipelineverarbeitung bezüglich des ersten Pixeldatenbusses, der zweiten Pixelverarbeitungseinrichtung, der Mehrzahl von dritten Pixelbussen, des Adreßbusses und der Mehrzahl von zweiten Verarbeitungseinrichtungen zur Bewegungsvoraussage umfaßt.

14. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 10, bei welcher  
die Pixelverarbeitungseinrichtung  
eine Mehrzahl von ersten Pixelverarbeitungseinrichtungen (34a, 34b) umfaßt, welche die durch den ersten  
60 Pixeldatenbus übertragenen Pixeldaten zum Gewinnen von Referenzdaten unterschiedlichen Typs zur Bewegungsvoraussage unter Verwendung der übertragenen Pixeldaten empfangen und welche ein drittes Datenport zum Ausgeben der Referenzdaten zur Bewegungsvoraussage und ein viertes Datenport zum Ausgeben der Referenzdaten zur Bewegungsvoraussage gemäß dem Typ der Referenzdaten zur Bewegungsvoraussage aufweisen;  
65 der zweite Pixeldatenbus eine Mehrzahl von dritten Pixeldatenbussen (LDB1, LDB2) umfaßt, welche der ersten Pixelverarbeitungseinrichtung entsprechend zum Übertragen der aus dem vierten Datenport ausgegebenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind;  
die zweite Pixeldaten-Speichereinrichtung eine Mehrzahl von dritten Pixeldaten-Speichereinrichtungen

- (56c, 56d) umfaßt, welche der ersten Pixelverarbeitungseinrichtung entsprechend zum Speichern der durch den dritten Pixeldatenbus übertragenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind; die Verarbeitungseinrichtung zur Bewegungsvoraussage eine Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage (34a, 34b) umfaßt, die der ersten Pixelverarbeitungseinrichtung entsprechend zum Ausführen der Bewegungsvoraussage unter Verwendung der Referenzdaten zur Bewegungsvoraussage vorgesehen sind, welche in der dritten Pixeldaten-Speichereinrichtung gespeichert sind und zum Erzeugen eines Bewegungsvektors durch den dritten Pixeldatenbus übertragen werden; und bei welcher
- die Pipelineverarbeitungseinrichtung eine Einrichtung (26) zum Steuern der Pipelineverarbeitung bezüglich des ersten Pixeldatenbusses, der Mehrzahl von ersten Pixelverarbeitungseinrichtungen, der Mehrzahl von dritten Pixeldatenbussen und der Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage umfaßt.
15. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 12, bei welcher die Mehrzahl der dritten Pixeldaten-Speichereinrichtungen unterschiedliche Referenzdaten zur Bewegungsvoraussage in Abhängigkeit davon speichert, ob die Daten für eine Helligkeitskomponente oder eine Farbdifferenzkomponente vorgesehen sind.
16. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 13, bei welcher die Mehrzahl von dritten Pixeldaten-Speichereinrichtungen unterschiedliche Referenzdaten zur Bewegungsvoraussage in Abhängigkeit davon speichert, ob die Daten für eine Helligkeitskomponente oder eine Farbdifferenzkomponente vorgesehen sind.
17. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 14, bei welcher die Mehrzahl von dritten Pixeldaten-Speichereinrichtungen unterschiedliche Referenzdaten zur Bewegungsvoraussage in Abhängigkeit davon speichert, ob die Daten für eine Helligkeitskomponente oder eine Farbdifferenzkomponente vorgesehen sind.
18. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 10, bei welcher die Pixelverarbeitungseinrichtung umfaßt:  
eine Bildverarbeitungseinrichtung (313, 314) zum Ausführen der diskreten Kosinsumwandlung und/oder der inversen diskreten Kosinsumwandlung sowie der Quantisierung und/oder der inversen Quantisierung zur Bilddatenkompression und/oder Bilddatenexpansion;  
wenigstens zwei Schnittstelleneinrichtungen (318a, 318b), die nur für Bilddaten vorgesehen sind;  
Speichereinrichtungen (331—333, 335) zum Speichern von Daten zur Bilddatenkompression und/oder Bilddatenexpansion, welche durch die Bildverarbeitungseinrichtung verarbeitet werden; und einen Datenbus (IPB2) zum Übertragen von Daten zwischen den Schnittstelleneinrichtungen und den Speichereinrichtungen.
19. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 10, bei welcher die Pixelverarbeitungseinrichtung umfaßt:  
eine Bildverarbeitungseinrichtung (313, 314) zum Ausführen der diskreten Kosinsumwandlung und/oder der inversen Kosinsumwandlung und der Quantisierung und/oder der inversen Quantisierung zur Bilddatenkompression und/oder Bilddatenexpansion; wenigstens zwei Schnittstelleneinrichtungen (318a, 318b), die nur für Bilddaten vorgesehen sind;  
Speichereinrichtungen (331—333, 335) zum Speichern von Daten zur Bilddatenkompression und/oder Bilddatenexpansion, welche durch die Bildverarbeitungseinrichtung verarbeitet werden; und eine den Schnittstelleneinrichtungen entsprechend vorgesehene Mehrzahl von Datenbussen (IPB2a, IPB2b) zum Übertragen von Daten zwischen jeder der Schnittstelleneinrichtungen und den Speichereinrichtungen.
20. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 3, bei welcher die Pixeldaten-Speichereinrichtung  $n$  ( $n \geq 2$ ) Speicherzellenanordnungen (801—832) zum Speichern der Pixeldaten zum Festlegen der Pixel auf einem Bildschirm umfaßt, wobei die  $n$  Speicherzellenanordnungen  $n$  Pixeldaten ausgeben, von denen jede in einer entsprechenden von den  $n$  Speicherzellenanordnungen in Reaktion auf das Anlegen von einem Zeilenadreßsignal und von einem Spaltenadreßsignal gespeichert wird, wobei die  $n$  Pixeldaten eine Mehrzahl von Pixeln festlegen, die in einer vertikalen Richtung auf dem Bildschirm angeordnet sind; und wobei die Einrichtung zur Steuerung der Pipelineverarbeitung ferner eine Einrichtung zur Adreßsignalerzeugung umfaßt, welche auf ein seriellcs Uhersignal zum seriellen Ausgeben von Spaltenadreßsignalen reagiert, um die Pixeldaten der in einer horizontalen Richtung auf dem Bildschirm angeordneten Pixel festzulegen.
21. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 3, bei welcher die Pixeldaten-Speichereinrichtung  $n$  ( $n \geq 2$ ) Speicherzellenanordnungen (801—832) zum Speichern der Pixeldaten zum Festlegen der Pixel auf einem Bildschirm umfaßt, wobei die  $n$  Speicherzellenanordnungen  $n$  Pixeldaten ausgeben, von denen jede in einer entsprechenden von den  $n$  Speicherzellenanordnungen in Reaktion auf das Anlegen von einem Zeilenadreßsignal und von einem Spaltenadreßsignal gespeichert wird, wobei die  $n$  Pixeldaten eine Mehrzahl von Pixeln festlegen, die in einer horizontalen Richtung auf dem Bildschirm angeordnet sind; und wobei die Einrichtung zur Steuerung der Pipelineverarbeitung ferner eine Einrichtung zur Adreßsignalerzeugung (840) umfaßt, welche auf ein seriellcs Uhersignal zum seriellen Ausgeben von Spaltenadreßsignalen reagiert, um die Pixeldaten der in einer vertikalen Richtung auf dem Bildschirm angeordneten Pixel festzulegen.
22. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 2, bei welcher die Einrichtung zur Verarbeitung einer variablen Länge ferner umfaßt:

eine Speichereinrichtung (481) zum Speichern eines Codes variabler Länge, der einer Kombination aus Laufdaten und Pegeldaten, aus deren Codelängendaten und aus einem Flag zur Verarbeitung eines Codes festgelegter Länge entspricht;

eine auf angelegte Laufdaten und angelegte Pegeldaten reagierende Leseeinrichtung (531, 561, 564, 534) zum Lesen des entsprechenden Codes variabler Länge, der Codelängendaten desselben und des Flags zur Verarbeitung eines Codes festgelegter Länge aus der Speichereinrichtung;

eine Einrichtung zum Codieren einer festgelegten Länge (527) zum Ausführen des Codierens an den angelegten Laufdaten und den angelegten Pegeldaten gemäß einer vorbestimmten Regel, um einen entsprechenden Code festgelegter Länge auszugeben;

eine auf das aus der Leseeinrichtung ausgegebene Flag zur Verarbeitung eines Codes festgelegter Länge reagierende Auswahlrichtung (530) zum Auswählen eines des Codes variabler Länge und des Codes festgelegter Länge; und

eine Codekoppleinrichtung (505) zum fortlaufenden Koppeln der aus der Auswahlrichtung ausgegebenen Codes.

23. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 2, bei welcher die Einrichtung zur Verarbeitung einer variablen Länge ferner umfaßt:

eine Speichereinrichtung (482) zum Speichern von einem Code variabler Länge entsprechenden Laufdaten sowie Pegeldaten und eines Flags zur Verarbeitung einer festgelegten Länge;

eine Gewinnungseinrichtung (505) zum Gewinnen eines Codes variabler Länge aus einer angelegten Codefolge variabler Länge, welcher gekoppelte Codes variabler Länge umfaßt;

eine Leseeinrichtung (502), die auf das aus der Gewinnungseinrichtung gewonnenen Code variabler Länge reagieren kann, zum Lesen der entsprechenden Laufdaten, Pegeldaten und des Flags zur Verarbeitung einer festgelegten Länge aus der Speichereinrichtung;

eine Einrichtung zum Decodieren einer festgelegten Länge (526) zum Ausführen des Decodierens gemäß einer vorbestimmten Regel an dem durch die Gewinnungseinrichtung gewonnenen Code variabler Länge und zum Ausgeben entsprechend decodierter Daten; und

eine Auswahlrichtung (530), die auf das aus der Leseeinrichtung ausgegebene Flag zur Verarbeitung eines Codes festgelegter Länge reagiert, um entweder die Laufdaten und die Pegeldaten oder die decodierten Daten auszuwählen.

24. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 1, bei welcher

die Steuereinrichtung ferner eine Einrichtung zur Steuerung des Parallelbetriebs (11) zum Steuern eines Parallelbetriebs von vorbestimmten Prozessen und von Bilddaten zur Bilddatenkompression und/oder Bilddatenexpansion umfaßt; wobei

die Vorrichtung zur Bilddatenverarbeitung ferner eine Mehrzahl von Einrichtungen zur Ausführungsanweisung (14—16) umfaßt, von denen jede auf ein aus der Einrichtung zur Steuerung des Parallelbetriebs angelegtes Steuersignal reagiert, um die Ausführung des vorbestimmten Prozesses und die Übertragung der Bilddaten unabhängig anzuweisen.

25. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 24, ferner umfassend:

einen ersten Datenbus (214) zum Übertragen der Daten zwischen der Einrichtung zur Steuerung des Parallelbetriebs und der Mehrzahl von Einrichtungen zur Ausführungsanweisung sowie einen zweiten Datenpfad (210—213) zum Übertragen der Daten zwischen jeder der Mehrzahl von Einrichtungen zur Ausführungsanweisung.

26. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 1, ferner umfassend:

eine erste Speichereinrichtung (7) zum Speichern von Tabellendaten zur Umwandlung;

eine zweite Speichereinrichtung (51) zum Speichern von zu verarbeitenden Bilddaten;

eine dritte Speichereinrichtung (81) zum Speichern von Bilddaten zur Anzeige;

eine vierte Speichereinrichtung (6), welche umgewandelte Bilddaten speichert;

eine erste Zugriffseinrichtung (17) zum Zugreifen auf die erste Speichereinrichtung;

eine zweite Zugriffseinrichtung (18) zum Zugreifen auf die zweite Speichereinrichtung;

eine dritte Zugriffseinrichtung (19) zum Zugreifen auf die dritte Speichereinrichtung und

eine vierte Zugriffseinrichtung (20) zum Zugreifen auf die vierte Speichereinrichtung; bei welcher

die Steuereinrichtung ferner eine Einrichtung zur Steuerung des Zugriffs (11) zum unabhängigen Steuern des Beginns des Zugriffsbetriebs der ersten bis vierten Zugriffseinrichtungen umfaßt.

27. Vorrichtung zur Bilddatenverarbeitung, umfassend:

einen Steuerprozessor (11) zum Steuern eines Parallelbetriebs bezüglich einer vorbestimmten Verarbeitung und zur Bilddatenübertragung für eine Bilddatenkompression und/oder Bilddatenexpansion und

eine Mehrzahl von Einrichtungen zur Ausführungsanweisung (14—16), von denen jede auf ein aus dem Steuerprozessor angelegtes Steuersignal reagiert, um die Ausführung der vorbestimmten Verarbeitung und die Bilddatenübertragung einzeln anzuweisen.

28. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 27, ferner umfassend:

einen ersten Datenbus (214) zum Übertragen der Daten zwischen dem Steuerprozessor und der Mehrzahl von Einrichtungen zur Ausführungsanweisung und

einen zweiten Datenpfad (210—213) zum Übertragen der Daten zwischen jeder der Mehrzahl von Einrichtungen zur Ausführungsanweisung.

29. Vorrichtung zur Bilddatenverarbeitung, welche dazu in der Lage ist, auf eine erste Speichereinrichtung (7) zum Speichern von Tabellendaten zur Umwandlung, auf eine zweite Speichereinrichtung (51) zum Speichern zu verarbeitender Bilddaten, auf eine dritte Speichereinrichtung (81) zum Speichern der Bilddaten zur Anzeige und auf eine vierte Speichereinrichtung (6) zum Speichern der umgewandelten Bilddaten

zuzugreifen, umfassend:

eine erste Zugriffseinrichtung (17) zum Zugreifen auf die erste Speichereinrichtung;  
 eine zweite Zugriffseinrichtung (18) zum Zugreifen auf die zweite Speichereinrichtung;  
 eine dritte Zugriffseinrichtung (19) zum Zugreifen auf die dritte Speichereinrichtung;  
 eine vierte Zugriffseinrichtung (20) zum Zugreifen auf die vierte Speichereinrichtung und  
 eine Einrichtung zur Steuerung des Zugriffs (11) zum unabhängigen Steuern des Beginns des Zugriffsbe- 5  
 triebs der ersten bis vierten Zugriffseinrichtungen.

30. Vorrichtung zur Bilddatenverarbeitung, umfassend:

eine Speichereinrichtung (481) zum Speichern eines Codes variabler Länge, welcher einer Kombination aus  
 Lauf- und Pegeldaten, aus deren Codelängendaten und aus einem Flag zur Verarbeitung eines Codes 10  
 festgelegter Länge entspricht;  
 eine auf angelegte Laufdaten und angelegte Pegeldaten reagierende Leseeinrichtung (531, 534, 561, 564)  
 zum Lesen eines entsprechenden Codes variabler Länge, der Codelängendaten desselben und eines Flags  
 zur Verarbeitung eines Codes festgelegter Länge aus der Speichereinrichtung;  
 eine Einrichtung zum Codieren mit festgelegter Länge (527) zum Ausführen des Codierens an den angeleg- 15  
 ten Laufdaten und den angelegten Pegeldaten gemäß einer vorbestimmten Regel, um einen entsprechenden  
 Code festgelegter Länge vorzusehen;  
 eine Auswahlrichtung (530), die auf das aus der Leseeinrichtung ausgegebene Flag zur Verarbeitung  
 eines Codes festgelegter Länge reagiert, um den Code variabler Länge und/oder den Code festgelegter  
 Länge auszuwählen; und 20  
 eine Codekoppleinrichtung (505) zum fortlaufenden Koppeln der aus der Auswahlrichtung ausgegebenen  
 Codes.

31. Vorrichtung zur Bilddatenverarbeitung, umfassend:

eine Speichereinrichtung (482) zum Speichern von Laufdaten und Pegeldaten, welche einem Code variabler  
 Länge und einem Flag eines Codes festgelegter Länge entsprechen; 25  
 eine Gewinnungseinrichtung (505) zum Gewinnen eines Codes variabler Länge aus einer angelegten  
 Codefolge variabler Länge, welcher gekoppelte Codes variabler Länge umfaßt;  
 eine Leseeinrichtung (502), die auf den durch die Gewinnungseinrichtung gewonnenen Code variabler  
 Länge reagiert, um die entsprechenden Laufdaten, die Pegeldaten und das Flag zur Verarbeitung einer  
 festgelegten Länge aus der Speichereinrichtung zu lesen; 30  
 eine Einrichtung zum Decodieren mit festgelegter Länge (526) zum Ausführen des Decodierens an dem  
 durch die Gewinnungseinrichtung gewonnenen Code variabler Länge gemäß einer vorbestimmten Regel,  
 um entsprechend decodierte Daten vorzusehen; und  
 eine Auswahlrichtung (530), welche dem aus der Leseeinrichtung ausgegebenen Flag zur Verarbeitung  
 eines Codes festgelegter Länge entspricht, um entweder die Laufdaten und Pegeldaten oder die decodierten 35  
 Daten auszuwählen.

32. Vorrichtung zur Bilddatenverarbeitung, umfassend:

$n$  ( $\geq 2$ ) Speicherzellenanordnungen (801—832) zum Speichern von Pixeldaten zum Festlegen der Pixel auf  
 einem Bildschirm, wobei  
 die  $n$  Speicherzellenanordnungen  $n$  Pixeldaten ausgeben, von denen jede in einer entsprechenden von den  $n$  40  
 Speicherzellenanordnungen in Reaktion auf das Anlegen von einem Zeilenadreßsignal und von einem  
 Spaltenadreßsignal gespeichert wird, wobei  
 die  $n$  Pixeldaten eine Mehrzahl von Pixeln festlegen, die in einer vertikalen Richtung auf dem Bildschirm  
 angeordnet sind; und  
 eine Einrichtung zur Adreßsignalerzeugung (840), welche auf ein serielles Uhrsignal zum seriellen Ausgeben 45  
 von Spaltenadreßsignalen reagiert, um die Pixeldaten der in einer horizontalen Richtung auf dem Bild-  
 schirm angeordneten Pixel festzulegen.

33. Vorrichtung zur Bilddatenverarbeitung, umfassend:

$n$  ( $\geq 2$ ) Speicherzellenanordnungen (801—832) zum Speichern von Pixeldaten zum Festlegen der Pixel auf  
 einem Bildschirm, wobei 50  
 die  $n$  Speicherzellenanordnungen  $n$  Pixeldaten ausgeben, von denen jede in einer entsprechenden von den  $n$   
 Speicherzellenanordnungen in Reaktion auf das Anlegen von einem Zeilenadreßsignal und von einem  
 Spaltenadreßsignal gespeichert wird, wobei  
 die  $n$  Pixeldaten eine Mehrzahl von Pixeln festlegen, die in einer horizontalen Richtung auf dem Bildschirm  
 angeordnet sind; und 55  
 eine Einrichtung zur Adreßsignalerzeugung (840), welche auf ein serielles Uhrsignal zum seriellen Ausgeben  
 von Spaltenadreßsignalen reagiert, um die Pixeldaten der in einer vertikalen Richtung auf dem Bildschirm  
 angeordneten Pixel festzulegen.

34. Vorrichtung zur Bilddatenverarbeitung, umfassend:

eine erste Pixeldaten-Speichereinrichtung (51) zum Speichern von zu komprimierenden Pixeldaten; 60  
 einen ersten Pixeldatenbus (PB) zum Übertragen der in der ersten Pixeldaten-Speichereinrichtung gespei-  
 cherten Pixeldaten;  
 eine Pixelverarbeitungseinrichtung (33), welche die durch den ersten Pixeldatenbus übertragenen Pixeldaten  
 empfängt, um unter Verwendung der übertragenen Pixeldaten Referenzdaten zur Bewegungsvoraussa-  
 ge zu gewinnen, und welche ein erstes und ein zweites Datenport (P1, P2) zum Ausgeben der Referenzdaten 65  
 zur Bewegungsvoraussage aufweist;  
 einen zweiten Pixeldatenbus (LDB) zum Übertragen der Referenzdaten zur Bewegungsvoraussage, welche  
 aus dem zweiten Datenport der Pixelverarbeitungseinrichtung ausgegeben werden;

eine zweite Pixeldaten-Speichereinrichtung (56) zum Speichern der durch den zweiten Datenbus übertragenen Referenzdaten zur Bewegungsvoraussage;

eine Verarbeitungseinrichtung zur Bewegungsvoraussage (47) zum Ausführen der Bewegungsvoraussage unter Verwendung der Referenzdaten zur Voraussage, welche in der zweiten Pixeldaten-Speichereinrichtung gespeichert sind und welche durch den zweiten Pixeldatenbus zum Erzeugen eines Bewegungsvektors übertragen werden; und

eine Einrichtung zum Steuern der Pipelineverarbeitung (23) zum Steuern der Pipelineverarbeitung bezüglich des ersten und des zweiten Pixeldatenbusses, der Pixelverarbeitungseinrichtung und der Verarbeitungseinrichtung zur Bewegungsvoraussage.

35. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 34, bei welcher die Einrichtung zum Steuern der Pipelineverarbeitung eine Systemsteuereinrichtung (114) zum Ausgeben eines Ausgangssteuersignals des ersten und des zweiten Datenports der Pixelverarbeitungseinrichtung umfaßt.

36. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 34, bei welcher die Pixelverarbeitungseinrichtung eine erste Pixelverarbeitungseinrichtung (34) zum Empfangen der durch den ersten Pixeldatenbus übertragenen Pixeldaten und zum Gewinnen der Referenzdaten zur Bewegungsvoraussage unter Verwendung der übertragenen Pixeldaten umfaßt, und welche ein drittes Datenport (P3) zum Ausgeben der Referenzdaten zur Bewegungsvoraussage sowie eine Mehrzahl von vierten Datenports (P4, P5) zum Verteilen und Ausgeben der Referenzdaten zur Bewegungsvoraussage aufweist;

der zweite Pixeldatenbus eine Mehrzahl von dritten Pixeldatenbussen (LDB1, LDB2) umfaßt, welche den vierten Datenports entsprechend zum Übertragen der aus den vierten Datenports ausgegebenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind;

die zweite Pixeldaten-Speichereinrichtung eine Mehrzahl von dritten Pixeldaten-Speichereinrichtungen (56a, 56b) umfaßt, welche den vierten Datenports entsprechend zum Speichern der durch den dritten Pixeldatenbus übertragenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind;

die Verarbeitungseinrichtung zur Bewegungsvoraussage eine Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage (47a, 47b) umfaßt, die den vierten Datenports entsprechend zum Ausführen der Bewegungsvoraussage unter Verwendung der Referenzdaten zur Bewegungsvoraussage vorgesehen sind, welche in der dritten Pixeldaten-Speichereinrichtung gespeichert sind und durch den dritten Pixeldatenbus übertragen werden, um einen Bewegungsvektor zu erzeugen; und bei welcher

die Einrichtung zur Steuerung der Pipelineverarbeitung eine Einrichtung (24) zum Steuern der Pipelineverarbeitung bezüglich des ersten Pixeldatenbusses, der ersten Pixelverarbeitungseinrichtung, der Mehrzahl von dritten Pixeldatenbussen und der Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage umfaßt.

37. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 36, welche ferner einen Adreßbus umfaßt, der mit der Mehrzahl von zweiten Pixeldaten-Speichereinrichtungen verbunden ist; bei welcher die erste Pixelverarbeitungseinrichtung eine zweite Pixelverarbeitungseinrichtung (34) zum Verteilen und Ausgeben der Referenzdaten zur Bewegungsvoraussage aus der Mehrzahl von vierten Datenports (P4, P5) in Abhängigkeit von Typen der Referenzdaten zur Bewegungsvoraussage umfaßt; wobei

die Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage eine Mehrzahl von zweiten Verarbeitungseinrichtungen zur Bewegungsvoraussage (47a, 47b) zum Erzeugen eines Bewegungsvektors in Abhängigkeit vom Typ der Referenzdaten zur Bewegungsvoraussage umfaßt; und wobei

die Einrichtung zur Steuerung der Pipelineverarbeitung eine Einrichtung (25) zum Steuern der Pipelineverarbeitung bezüglich des ersten Pixeldatenbusses, der zweiten Pixelverarbeitungseinrichtung, der Mehrzahl von dritten Pixeldatenbussen, des Adreßbusses und der Mehrzahl von zweiten Verarbeitungseinrichtungen zur Bewegungsvoraussage umfaßt.

38. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 34, bei welcher die Pixelverarbeitungseinrichtung

eine Mehrzahl von ersten Pixelverarbeitungseinrichtungen (34a, 34b) umfaßt, welche die durch den ersten Pixeldatenbus übertragenen Pixeldaten zum Gewinnen von Referenzdaten zur Bewegungsvoraussage unterschiedlichen Typs unter Verwendung der übertragenen Pixeldaten empfangen und welche ein drittes Datenport zum Ausgeben der Referenzdaten zur Bewegungsvoraussage und ein viertes Datenport zum Ausgeben der Referenzdaten zur Bewegungsvoraussage gemäß dem Typ der Referenzdaten zur Bewegungsvoraussage aufweisen; wobei

der zweite Pixeldatenbus eine Mehrzahl von dritten Pixeldatenbussen umfaßt, welche der ersten Pixelverarbeitungseinrichtung entsprechend zum Übertragen der aus dem vierten Datenport ausgegebenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind;

die zweite Pixeldaten-Speichereinrichtung eine Mehrzahl von dritten Pixeldaten-Speichereinrichtungen (56c, 56d) umfaßt, welche der ersten Pixelverarbeitungseinrichtung entsprechend zum Speichern der durch den dritten Pixeldatenbus übertragenen Referenzdaten zur Bewegungsvoraussage vorgesehen sind; die Verarbeitungseinrichtung zur Bewegungsvoraussage eine Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage (34a, 34b) umfaßt, die der ersten Pixelverarbeitungseinrichtung entsprechend zum Ausführen der Bewegungsvoraussage unter Verwendung der Referenzdaten zur Bewegungsvoraussage vorgesehen sind, welche in der dritten Pixeldaten-Speichereinrichtung gespeichert sind und durch den dritten Pixeldatenbus übertragen werden; und wobei

die Einrichtung zur Steuerung der Pipelineverarbeitung eine Einrichtung zum Steuern der Pipelineverarbeitung bezüglich des ersten Pixeldatenbusses, der Mehrzahl von ersten Pixelverarbeitungseinrichtungen,

der Mehrzahl von dritten Pixeldatenbussen und der Mehrzahl von ersten Verarbeitungseinrichtungen zur Bewegungsvoraussage umfaßt.

39. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 36, bei welcher die Mehrzahl von dritten Pixeldaten-Speichereinrichtungen unterschiedliche Referenzdaten zur Bewegungsvoraussage in Abhängigkeit davon speichert, ob die Daten für eine Helligkeitskomponente oder eine Farbdifferenzkomponente vorgesehen sind. 5

40. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 37, bei welcher die Mehrzahl der dritten Pixeldaten- Speichereinrichtungen unterschiedliche Referenzdaten zur Bewegungsvoraussage in Abhängigkeit davon speichert, ob die Daten für eine Helligkeitskomponente oder eine Farbdifferenzkomponente vorgesehen sind. 10

41. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 38, bei welcher die Mehrzahl der dritten Pixeldaten- Speichereinrichtungen unterschiedliche Referenzdaten zur Bewegungsvoraussage in Abhängigkeit davon speichert, ob die Daten für eine Helligkeitskomponente oder eine Farbdifferenzkomponente vorgesehen sind.

42. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 34, bei welcher die Pixelverarbeitungseinrichtung umfaßt:  
eine Bildverarbeitungseinrichtung (313, 314) zum Ausführen einer diskreten Kosinustransformation und/oder einer inversen diskreten Kosinustransformation sowie einer Quantisierung und/oder einer inversen Quantisierung zur Bilddatenkompression und/oder Bilddatenexpansion;  
wenigstens zwei Schnittstelleneinrichtungen (318a, 318b), die nur für die Bilddaten vorgesehen sind;  
Speichereinrichtungen (331—333, 335) zum Speichern der Daten zur Bilddatenkompression und/oder Bilddatenexpansion, welche durch die Bildverarbeitungseinrichtung verarbeitet werden; und  
einen Datenbus (IPB2) zum Übertragen der Daten zwischen den Schnittstelleneinrichtungen und den Speichereinrichtungen. 15 20

43. Vorrichtung zur Bilddatenverarbeitung nach Anspruch 34, bei welcher die Pixelverarbeitungseinrichtung umfaßt:  
eine Bildverarbeitungseinrichtung (313, 314) zum Ausführen einer diskreten Kosinustransformation und/oder einer inversen diskreten Kosinustransformation und einer Quantisierung und/oder einer inversen Quantisierung zur Bilddatenkompression und/oder Bilddatenexpansion;  
wenigstens zwei Schnittstelleneinrichtungen (318a, 318b), die nur für die Bilddaten vorgesehen sind;  
Speichereinrichtungen (331—333, 335) zum Speichern der Daten zur Bilddatenkompression und/oder Bilddatenexpansion, welche durch die Bildverarbeitungseinrichtung verarbeitet werden; und  
eine den Schnittstelleneinrichtungen entsprechend vorgesehene Mehrzahl von Datenbussen (IPB2a, IPB2b) zum Übertragen der Daten zwischen jeder der Schnittstelleneinrichtungen und den Speichereinrichtungen. 25 30

44. Verfahren zur Verarbeitung von Bilddaten zum Codieren einer Mehrzahl von Makroblockdaten, welche Pixel festlegen, die bei einer Pipelineverarbeitung eine Bildebene bilden, umfassend:  
einen ersten Schritt zum Ausführen einer diskreten Kosinustransformation und einer Quantisierung an den Makroblockdaten, um einen Pixelcode zu erzeugen (S7);  
einen zweiten Schritt zum Ausführen eines Kodierens mit variabler Länge an dem im ersten Schritt erzeugten Pixelcode, um einen Code variabler Länge zu erzeugen (S9); und  
einen dritten Schritt zum Koppeln des im zweiten Schritt erzeugten Codes variabler Länge, um eine Codefolge variabler Länge zu erzeugen. 35 40

45. Verfahren zur Verarbeitung von Bilddaten zum Decodieren einer Codefolge variabler Länge, die eine Mehrzahl von Codes variabler Länge umfaßt, welche Pixel festlegen, die bei einer Pipelineverarbeitung eine Bildebene bilden, umfassend:  
einen ersten Schritt zum Gewinnen des Codes variabler Länge aus der Codefolge variabler Länge (632);  
einen zweiten Schritt zum Ausführen eines Decodierens mit variabler Länge an dem im ersten Schritt gewonnenen Code variabler Länge, um einen Pixelcode zu erzeugen; und  
einen dritten Schritt zum Ausführen einer inversen Quantisierung und einer inversen diskreten Kosinustransformation an dem im zweiten Schritt erzeugten Pixelcode, um Makroblockdaten zu erzeugen, welche die die Bildebene bildenden Pixel festlegen. 45 50

46. Verfahren zur Verarbeitung von Bilddaten zum Abbilden von ersten Pixeldaten auf eine Speichereinrichtung (51), welche eine Mehrzahl der ersten Pixeldaten speichert, die Pixel auf einem Bildschirm festlegen, umfassend:  
einen ersten Schritt zum Bestimmen einer Mehrzahl von zweiten Pixeldaten, die zweidimensional zueinander benachbart sind und in einem rechteckigen Gebiet (861) aus den ersten Pixeldaten enthalten sind;  
einen zweiten Schritt zum Zuweisen von einer Adresse der Speichereinrichtung an die zweiten Pixeldaten, wobei die zweiten Pixeldaten als ein Datenstück betrachtet werden; und einen dritten Schritt zum Übertragen der zweiten Daten aus oder zu der Speichereinrichtung unter Verwendung der Adresse der zweiten Pixeldaten. 55 60

47. Verfahren zur Verarbeitung von Bilddaten zum Abbilden von ersten und zweiten Farbdifferenz-Pixeldaten auf eine Speichereinrichtung (51), welche die ersten und die zweiten Farbdifferenz-Pixeldaten speichert, die zum Festlegen von Pixeln auf einem Bildschirm miteinander in Beziehung stehen, umfassend:  
einen ersten Schritt zum abwechselnden Abbilden der ersten und der zweiten Farbdifferenz-Pixeldaten auf die Speichereinrichtung durch eine Übertragungseinheit (868); und  
einen zweiten Schritt zum Übertragen der ersten oder der zweiten Farbdifferenz-Pixeldaten aus oder zu der Speichereinrichtung durch die Übertragungseinheit. 65

48. Verfahren zur Verarbeitung von Bilddaten zum Abbilden von ersten und zweiten Farbdifferenz-Pixeldaten

ten auf eine Speichereinrichtung (51), welche die ersten und die zweiten Farbdifferenz-Pixeln speichert, die zum Festlegen von Pixeln auf einem Bildschirm miteinander in Beziehung stehen, umfassend:  
einen ersten Schritt zum Abbilden von dritten Farbdifferenz-Pixeln (880, 881), welche die gleiche Anzahl der ersten und der zweiten Farbdifferenz-Pixeln umfassen, als ein Datenstück auf die Speichereinrichtung und  
einen zweiten Schritt zum Übertragen der dritten Farbdifferenz-Pixeln als ein Datenstück aus oder zu der Speichereinrichtung.

Hierzu 82 Seite(n) Zeichnungen



FIG. 1

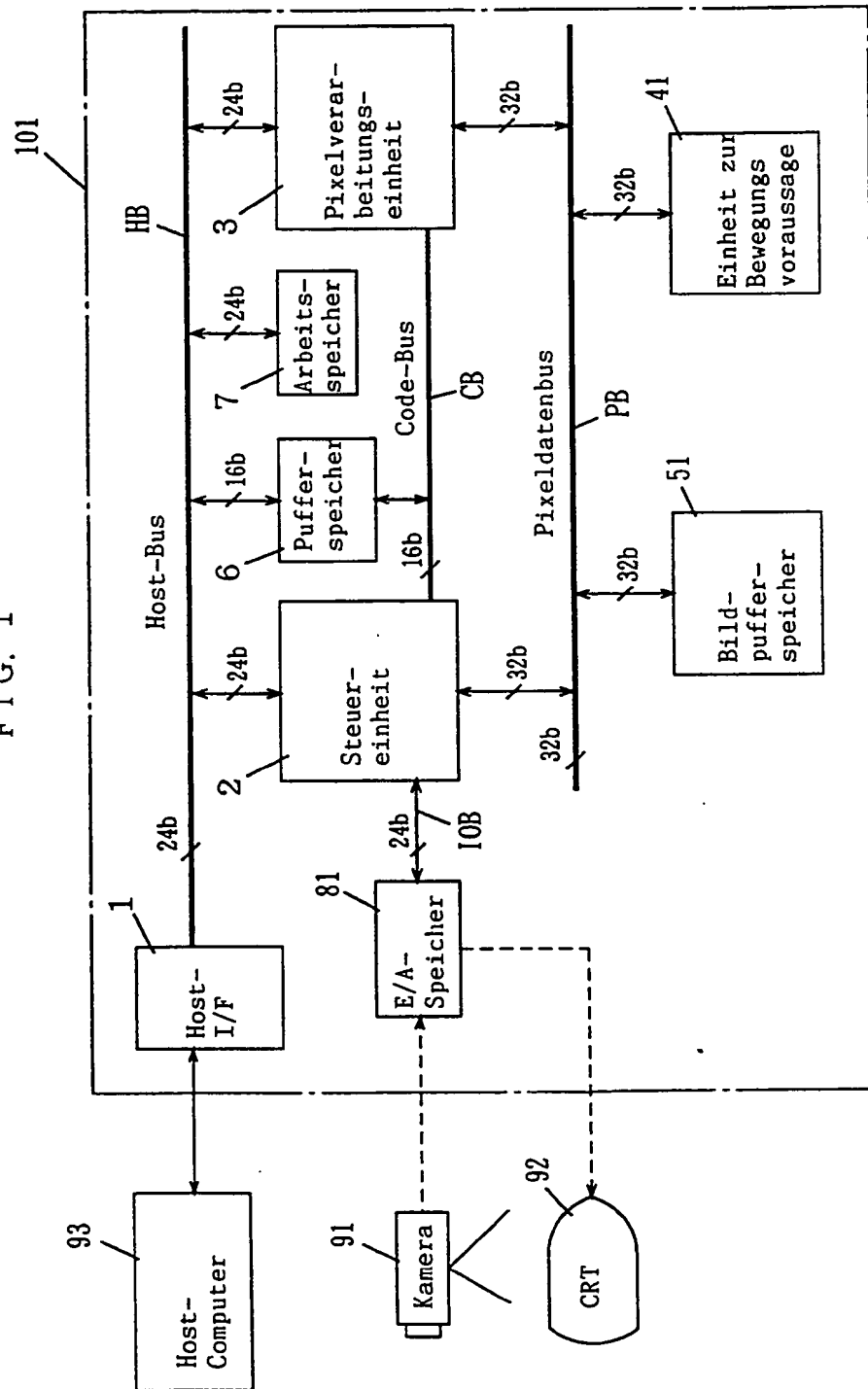


FIG. 2

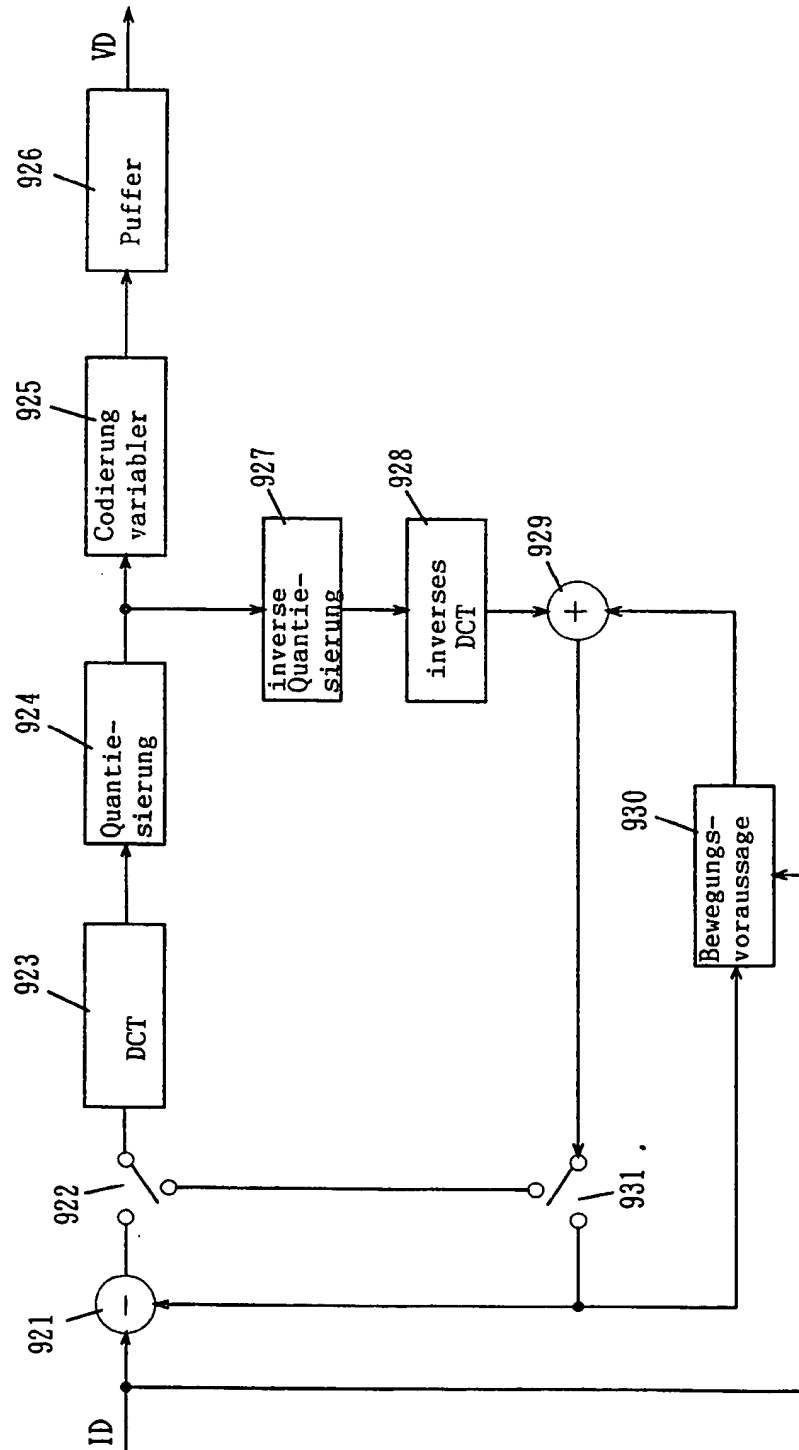


FIG. 3

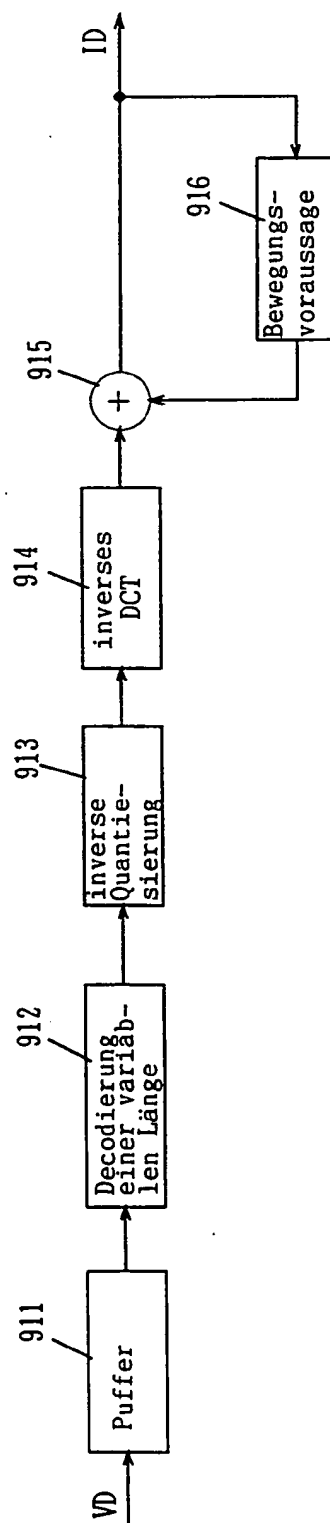
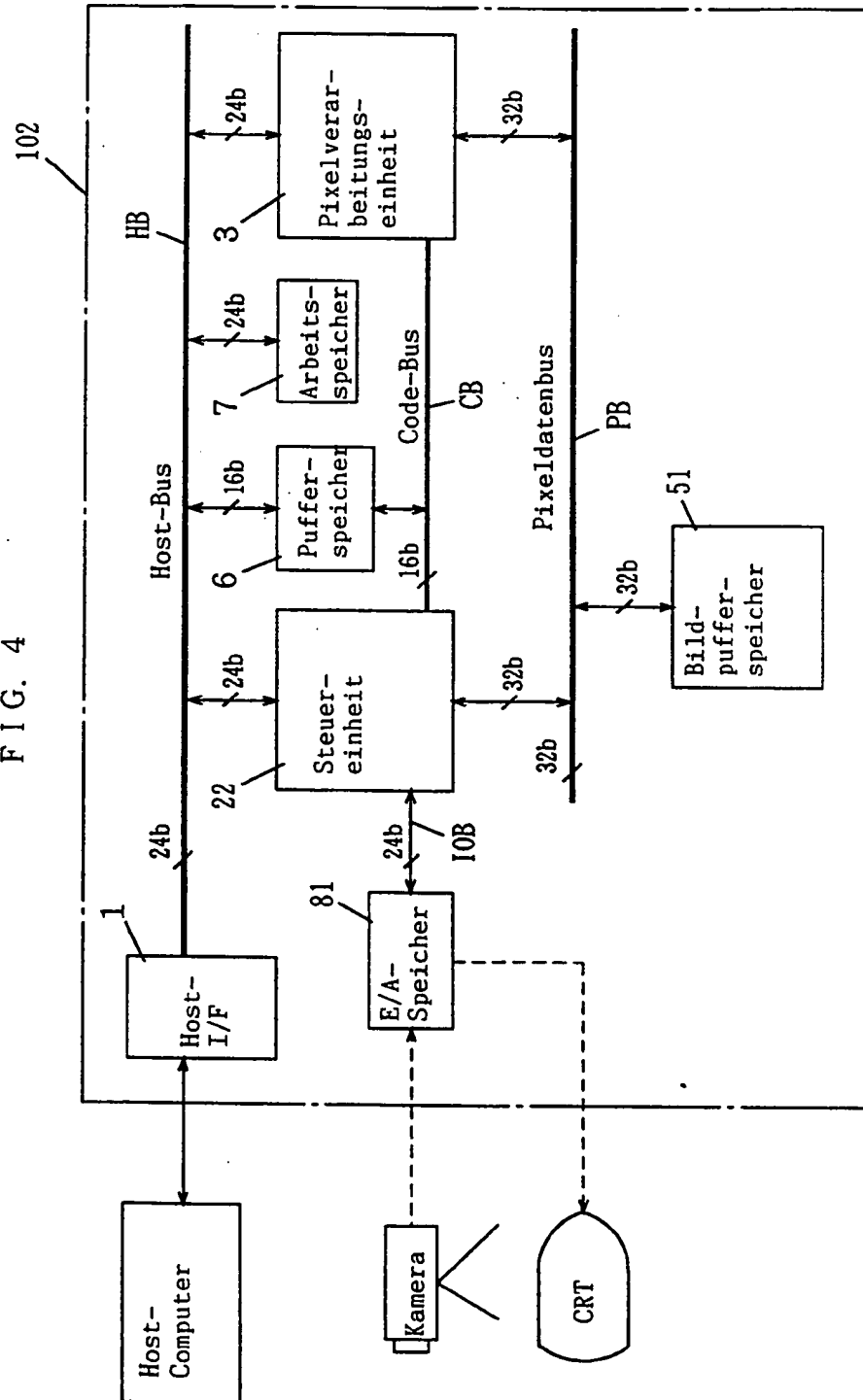


FIG. 4



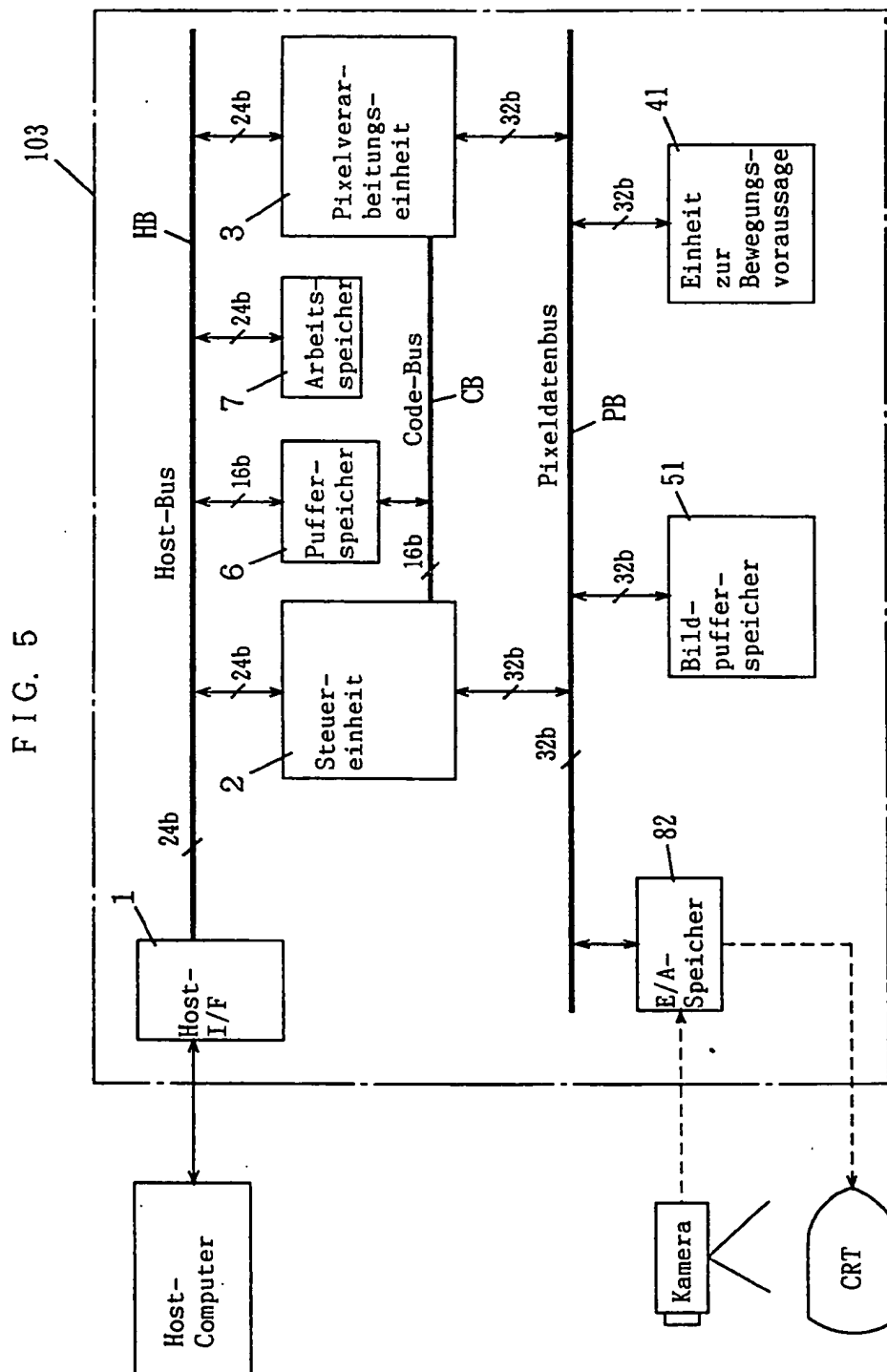


FIG. 6

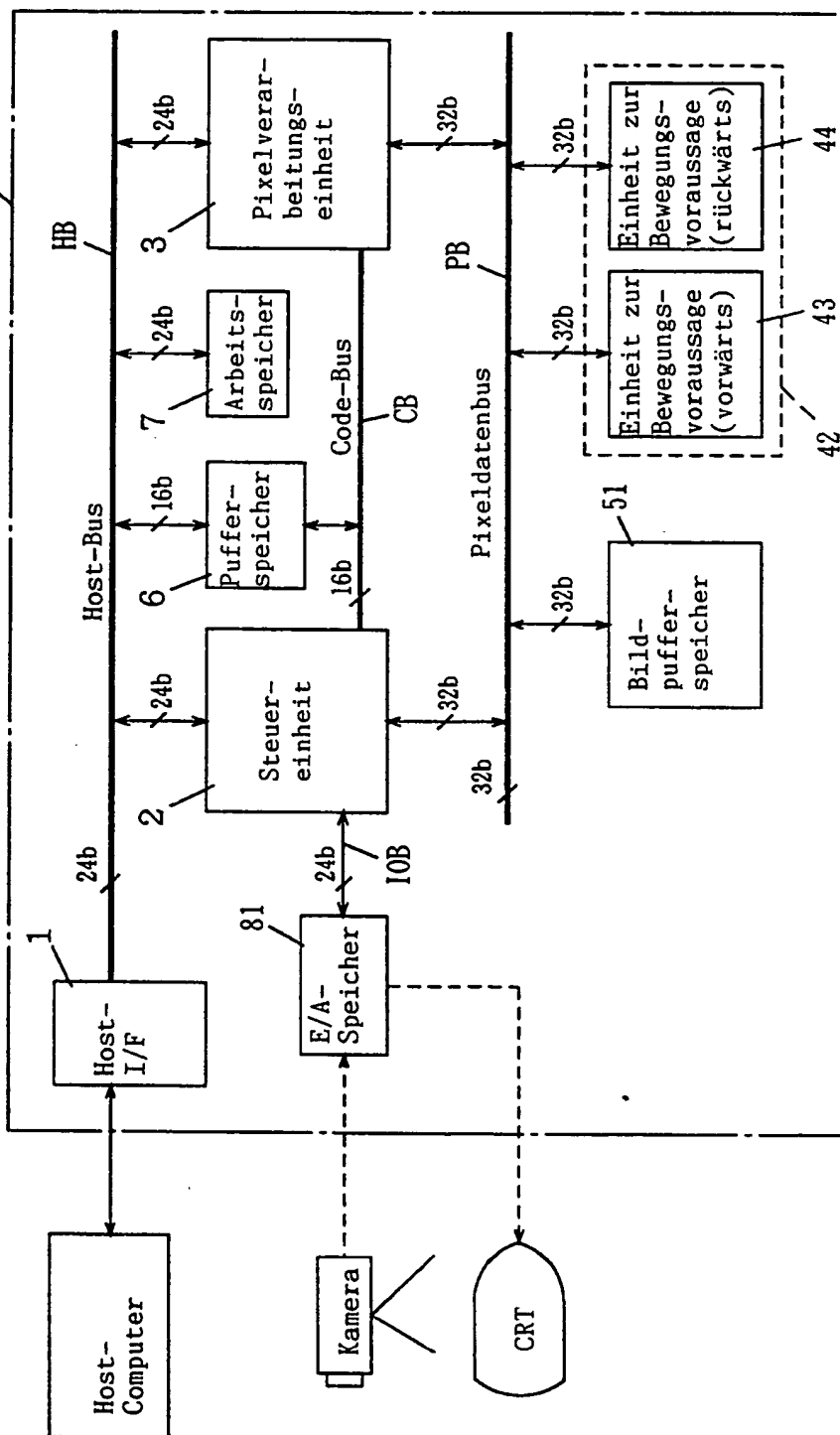


FIG. 7

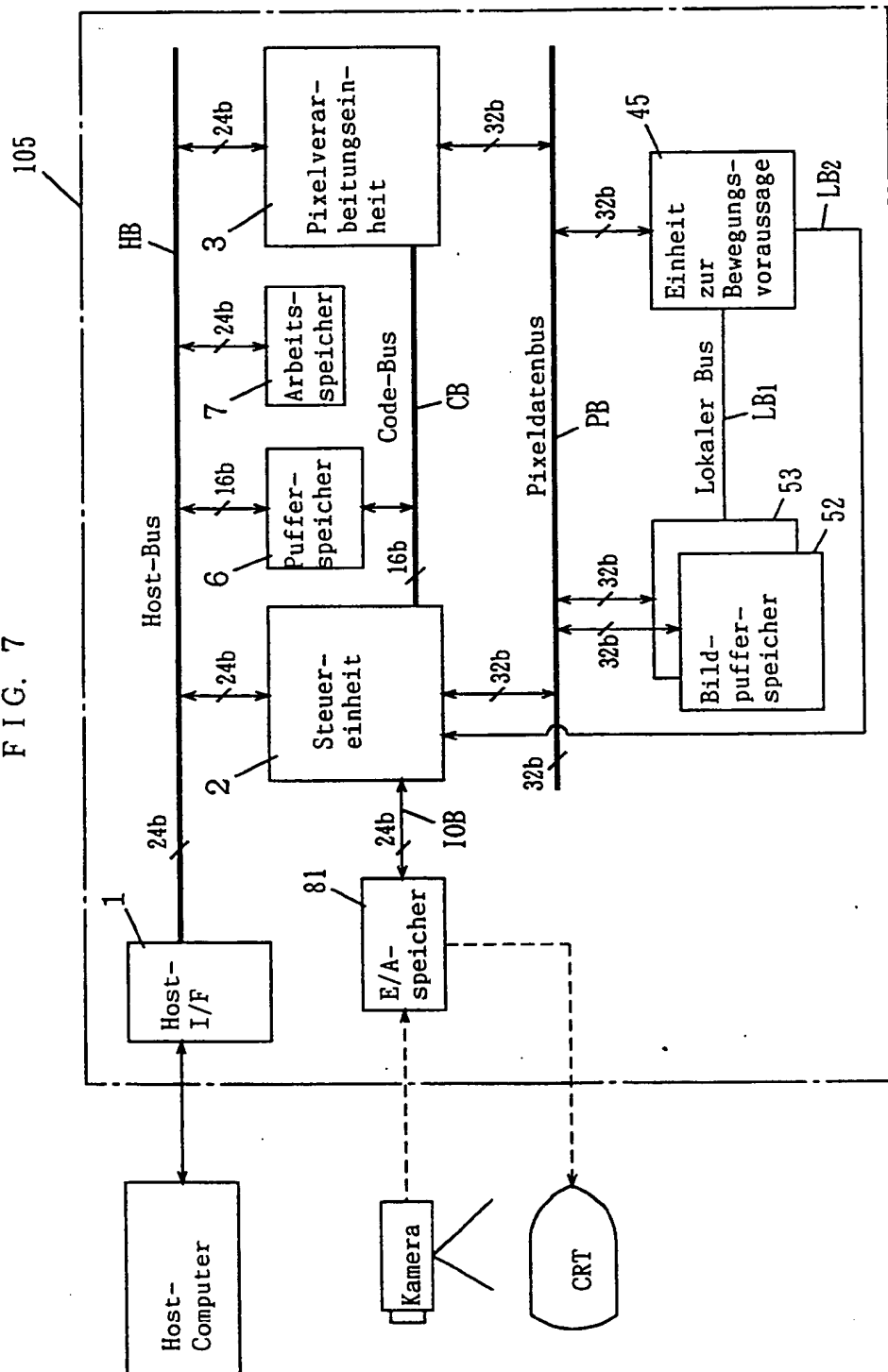




FIG. 8

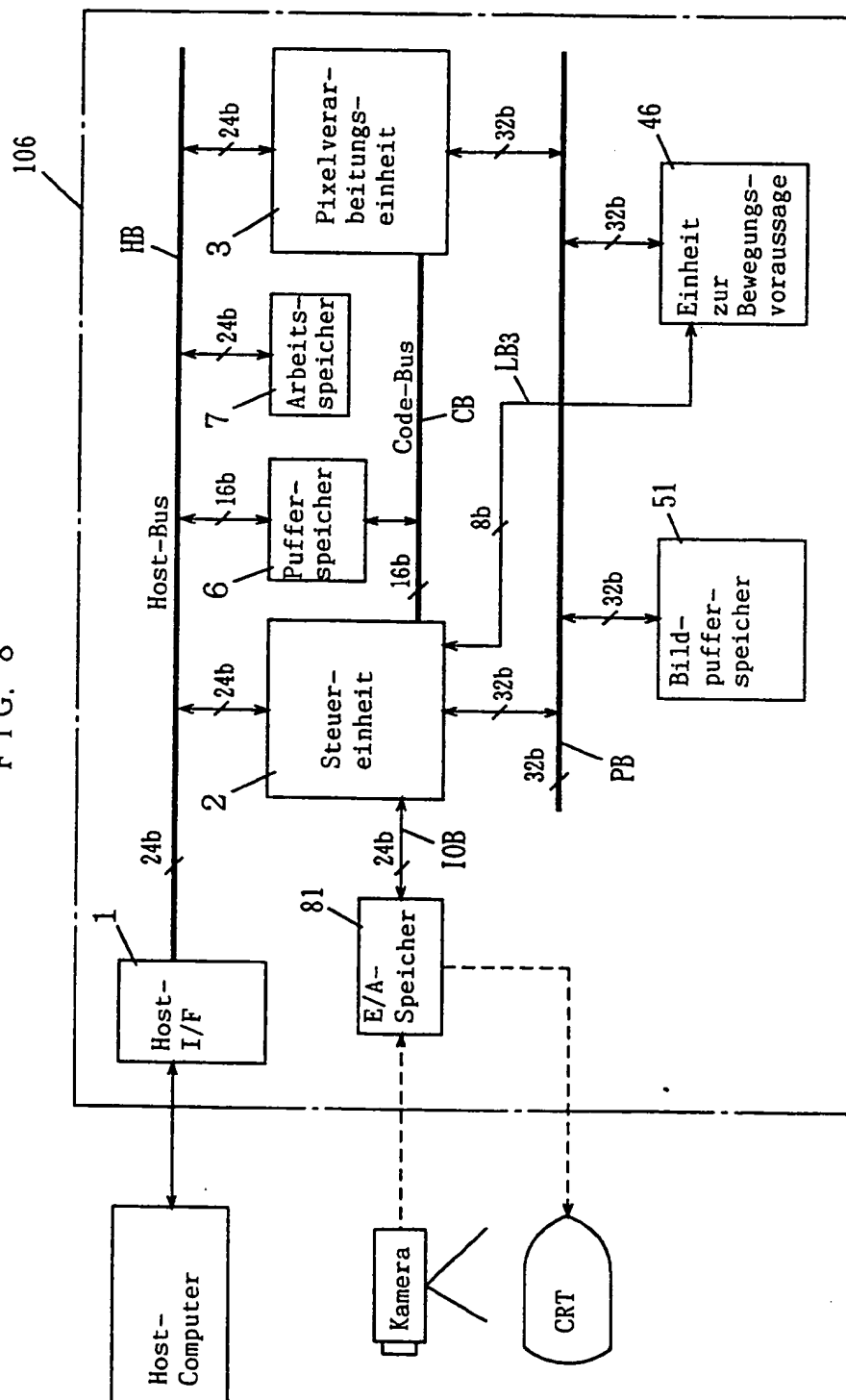


FIG. 9

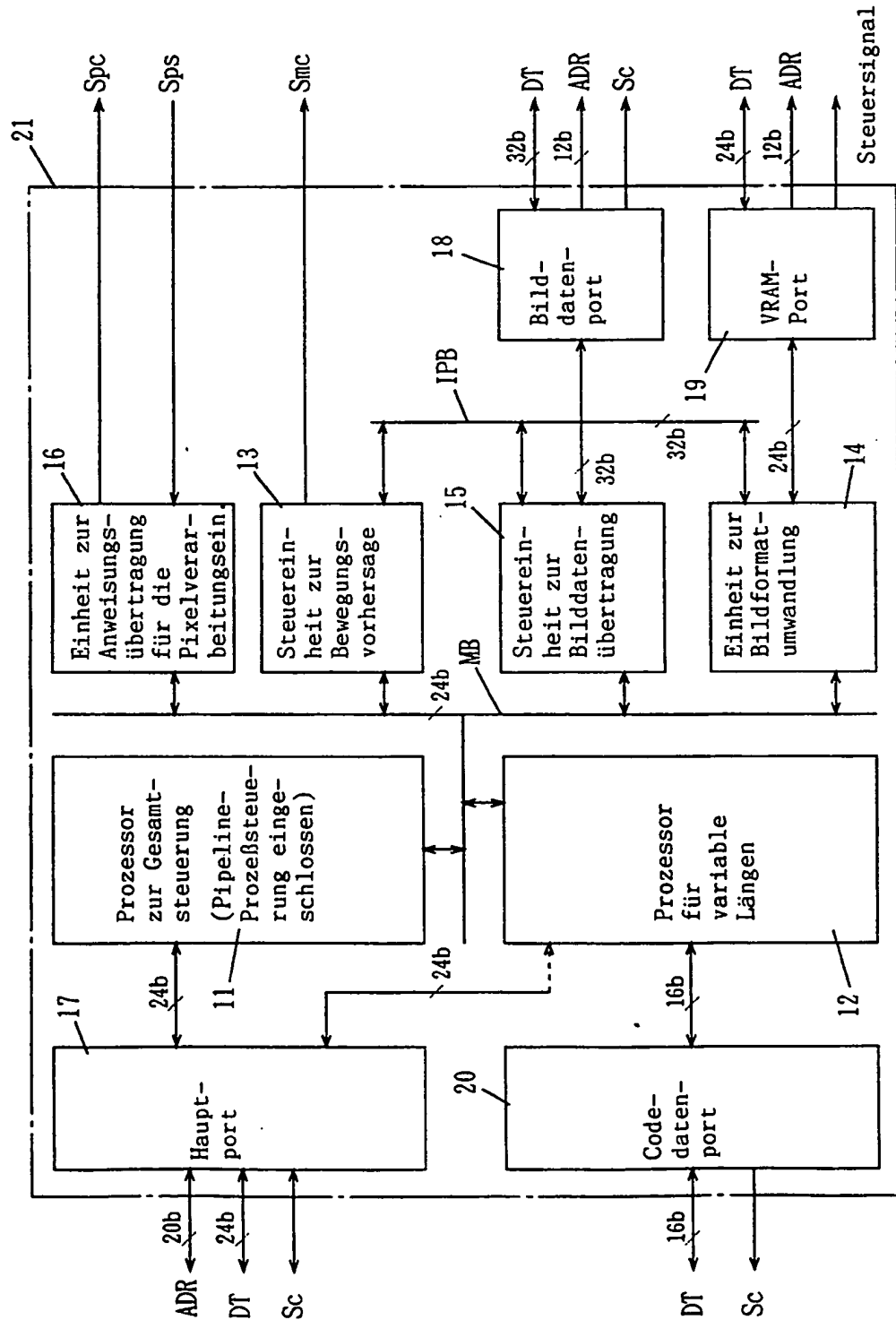


FIG. 10

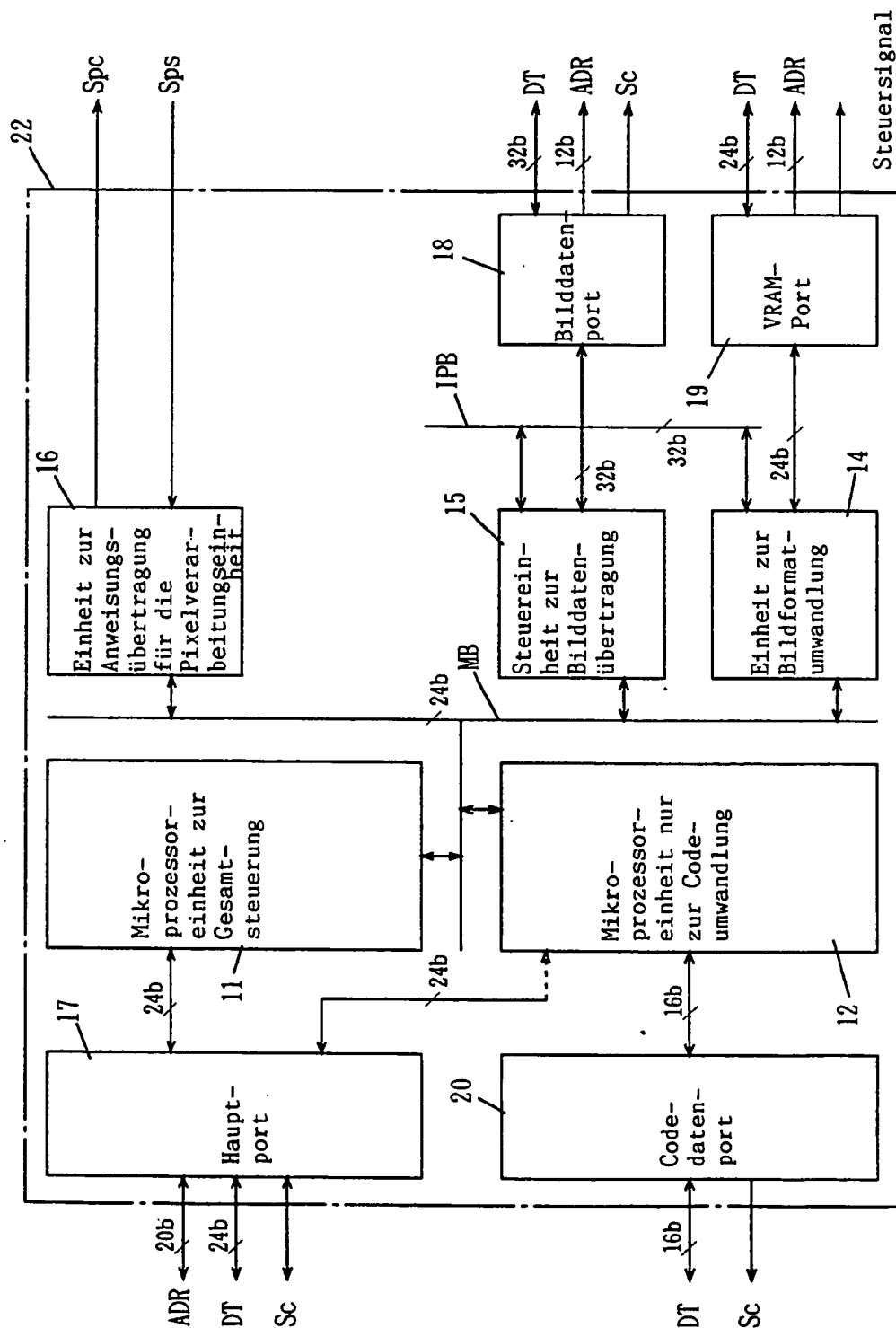


FIG. 11

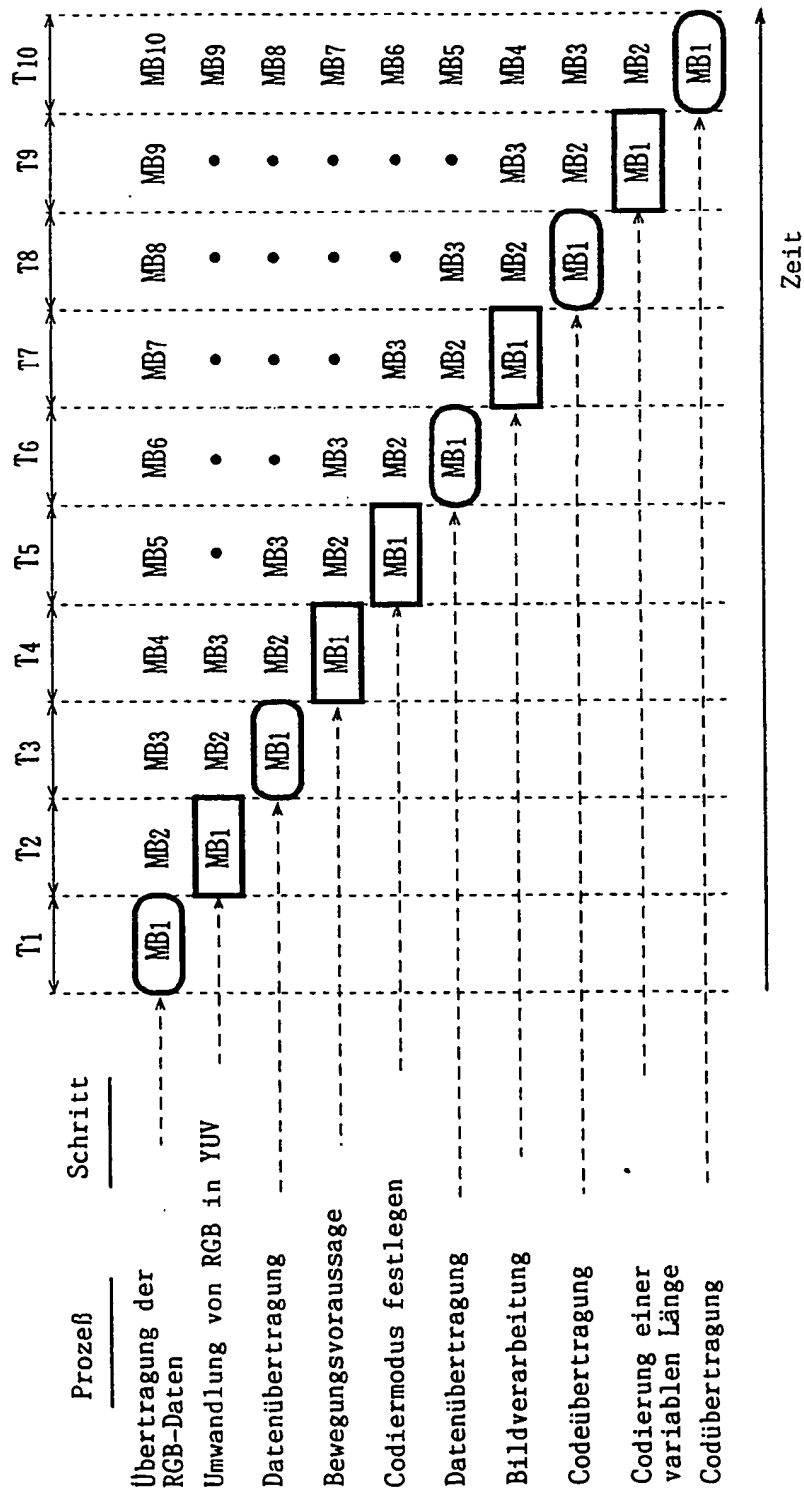


FIG. 12

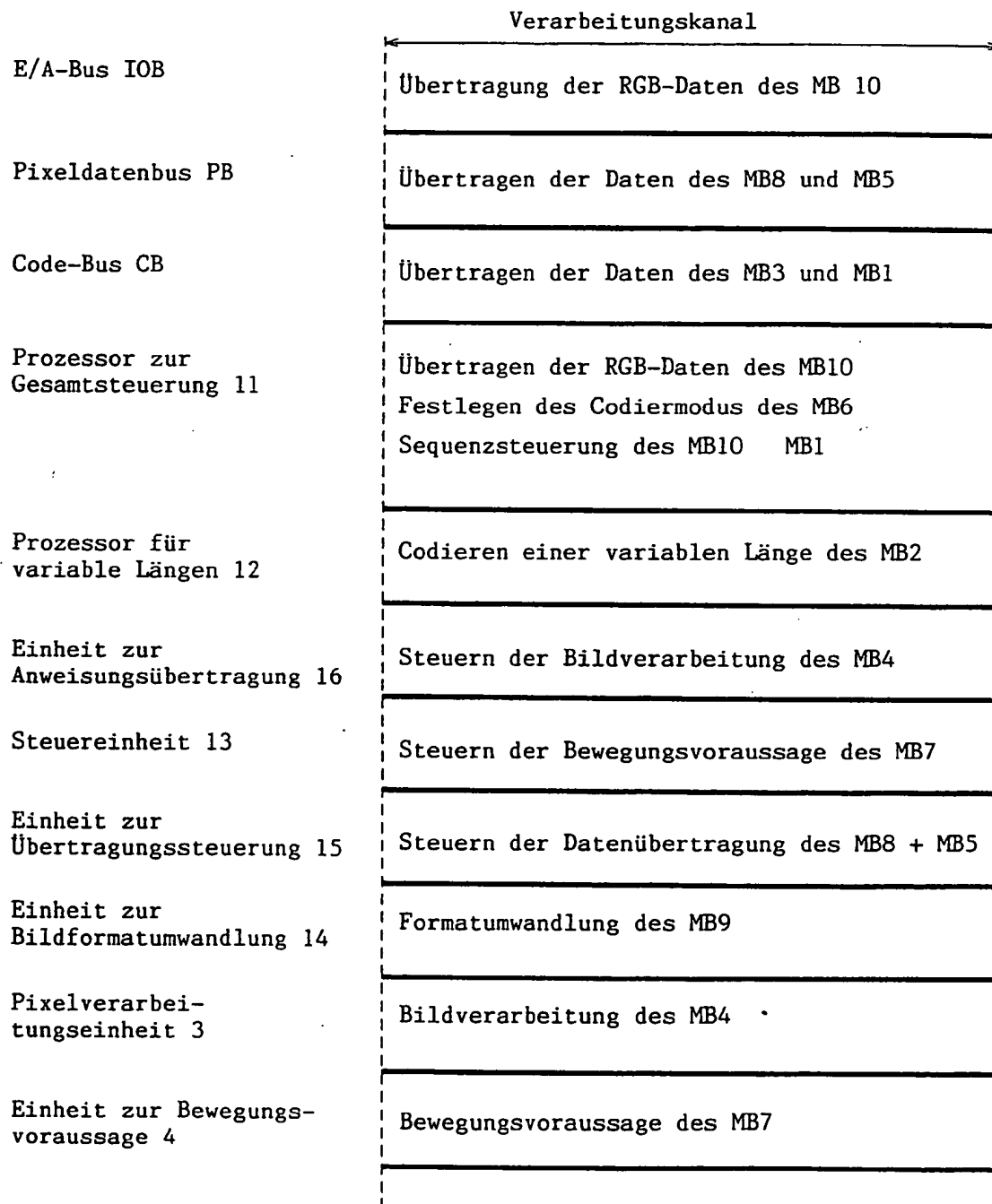


FIG. 13

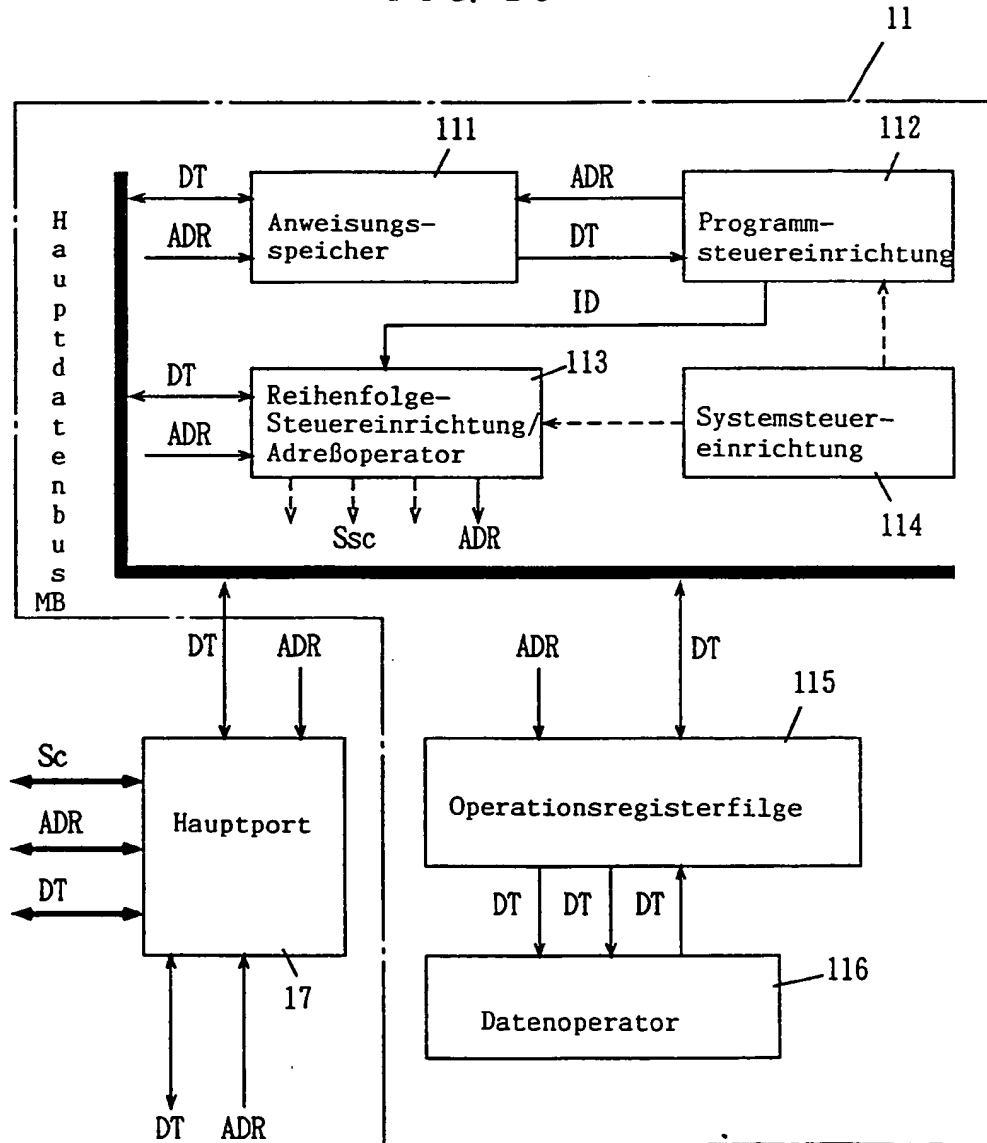


FIG. 14

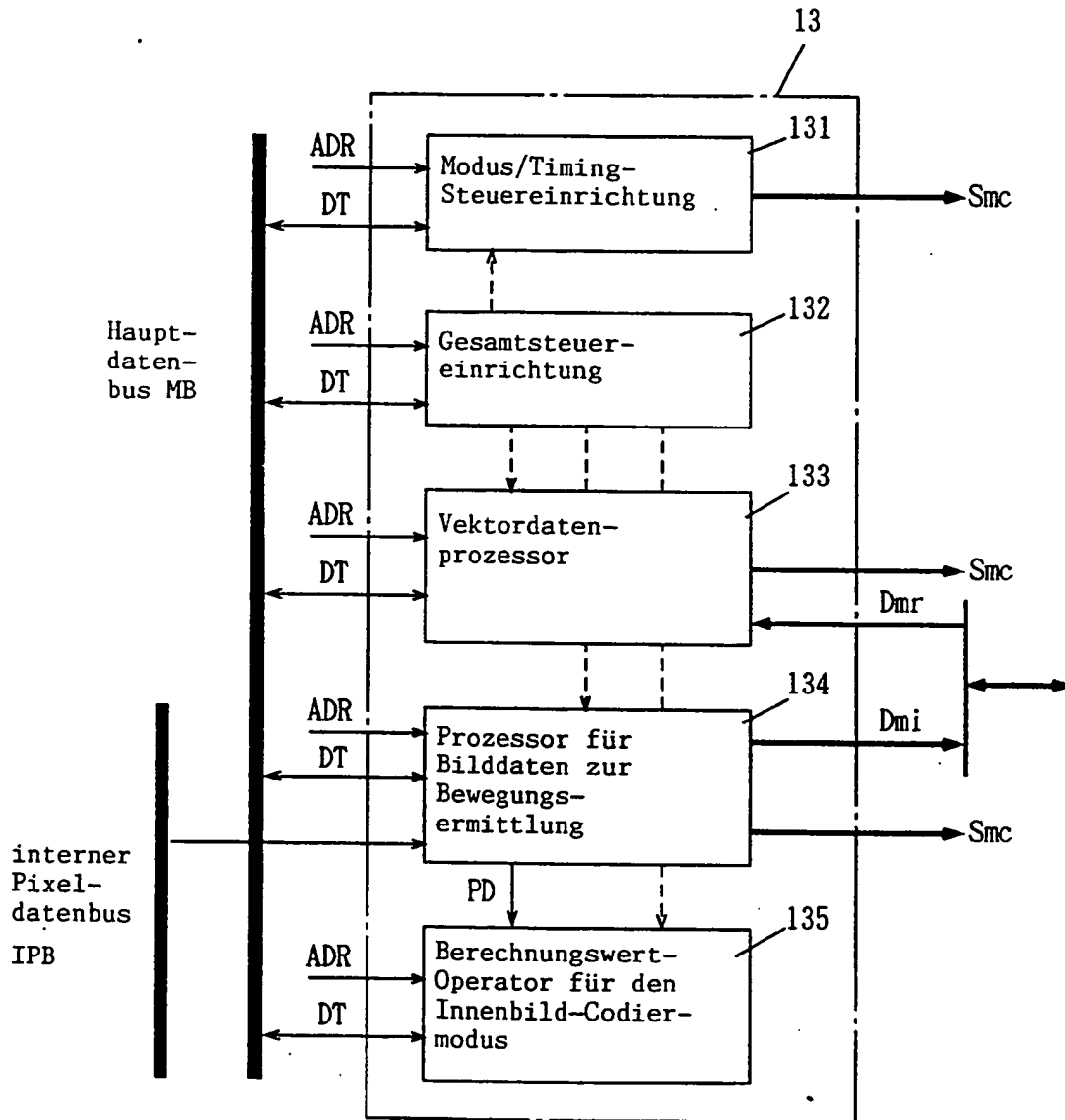




FIG. 15

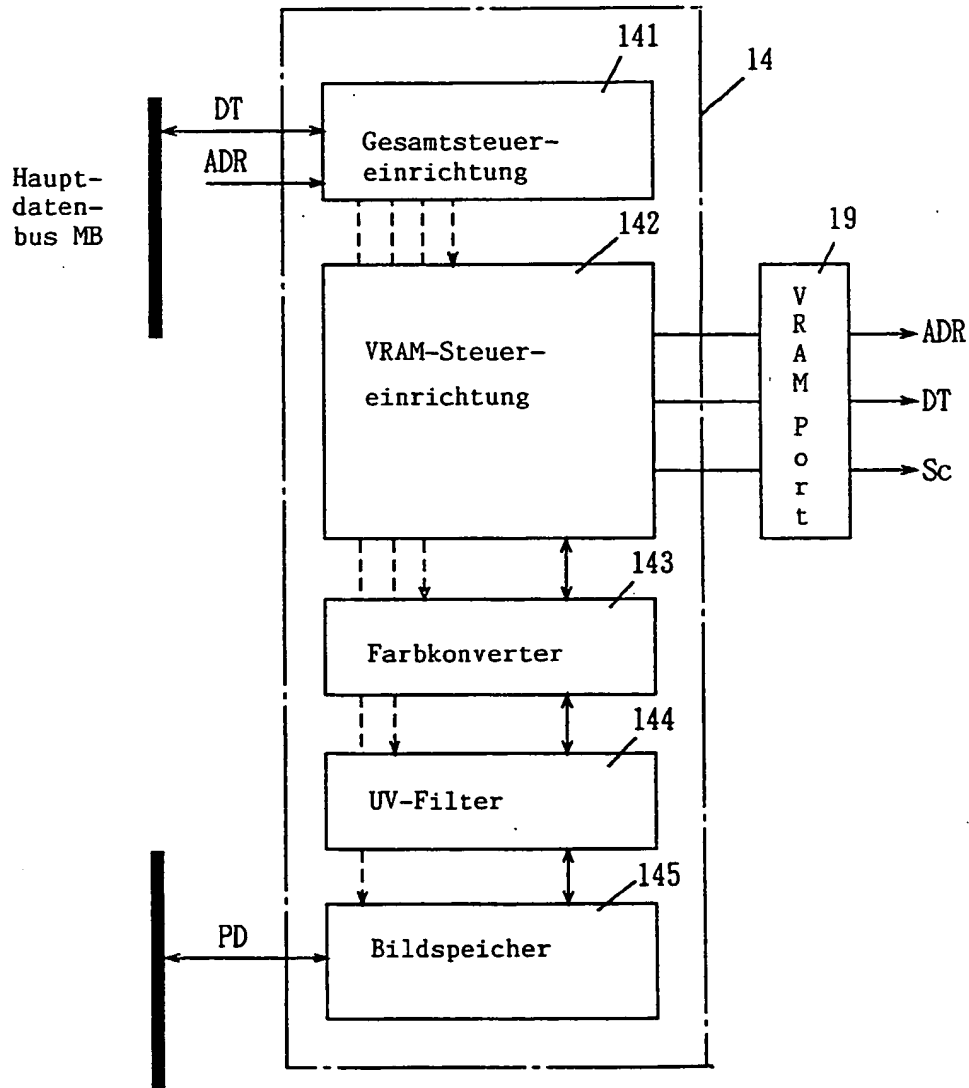


FIG. 16

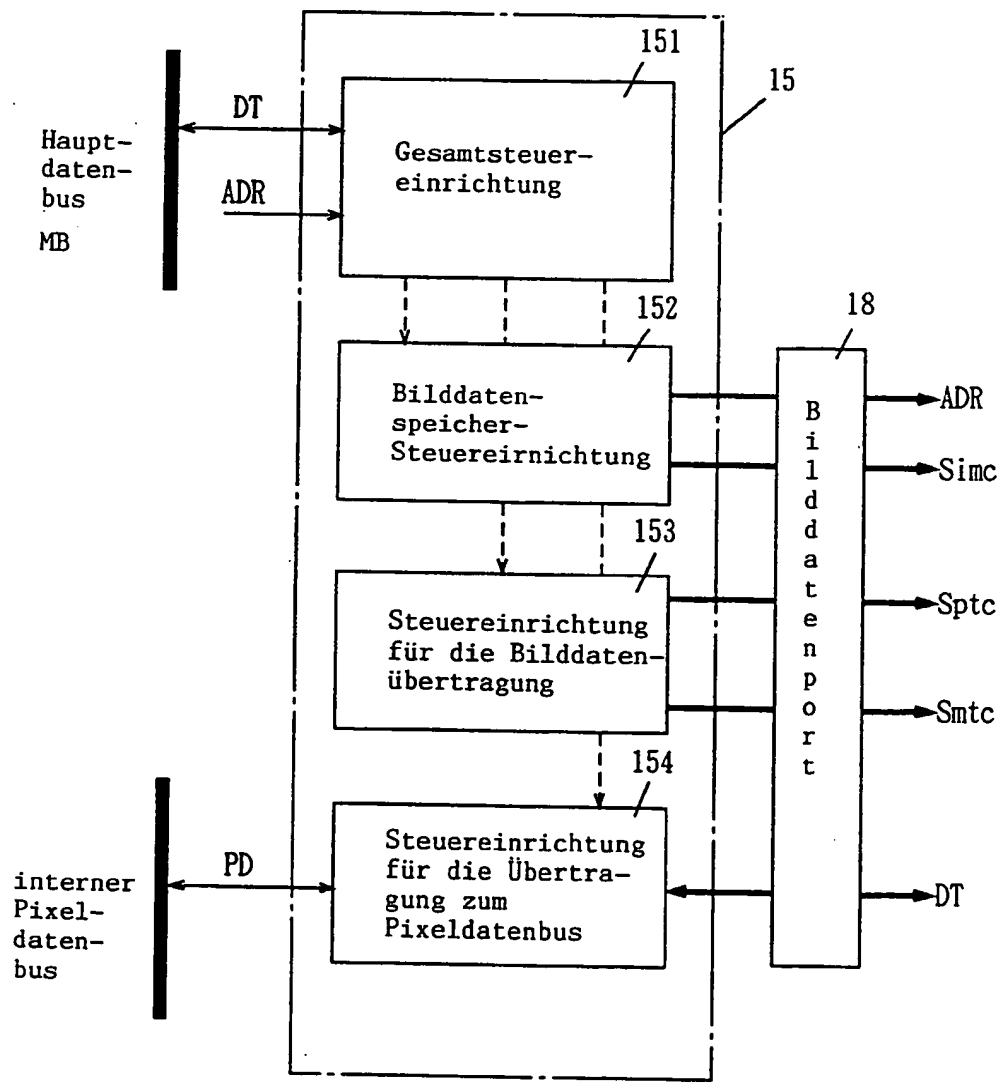


FIG. 17

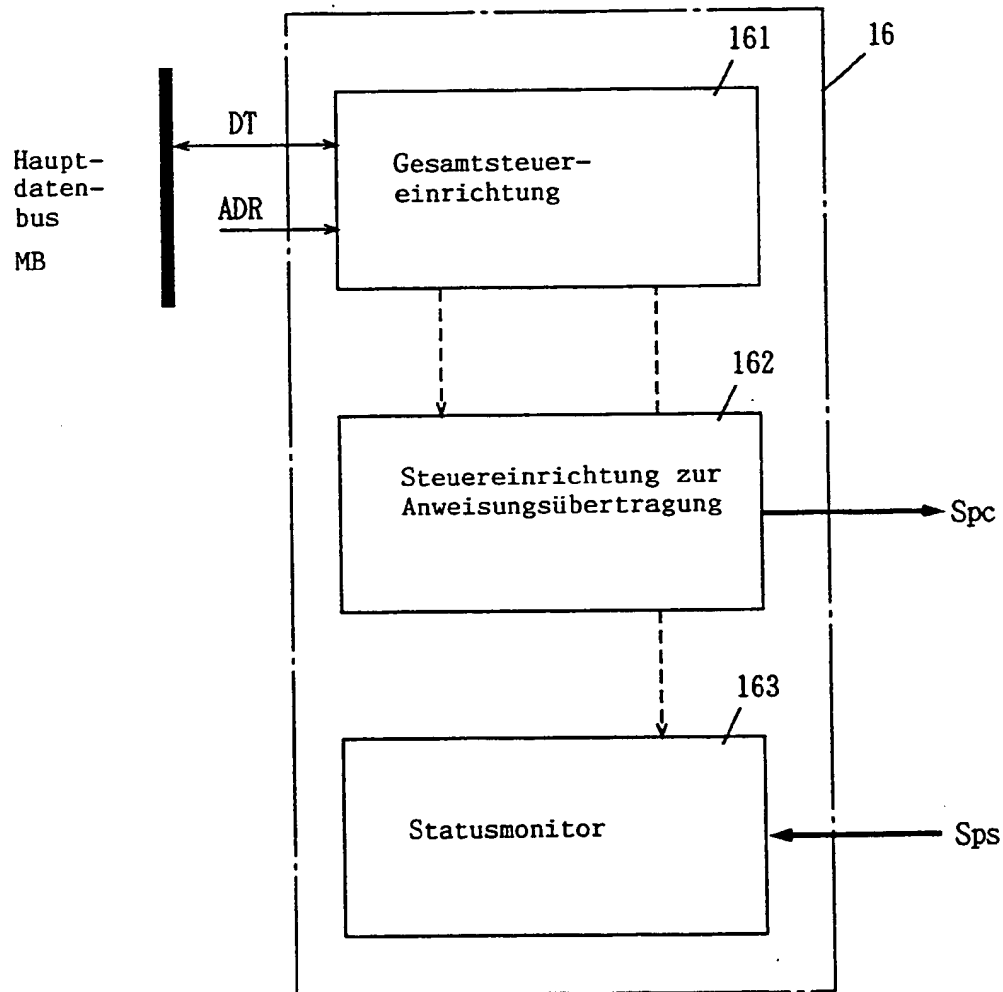


FIG. 18

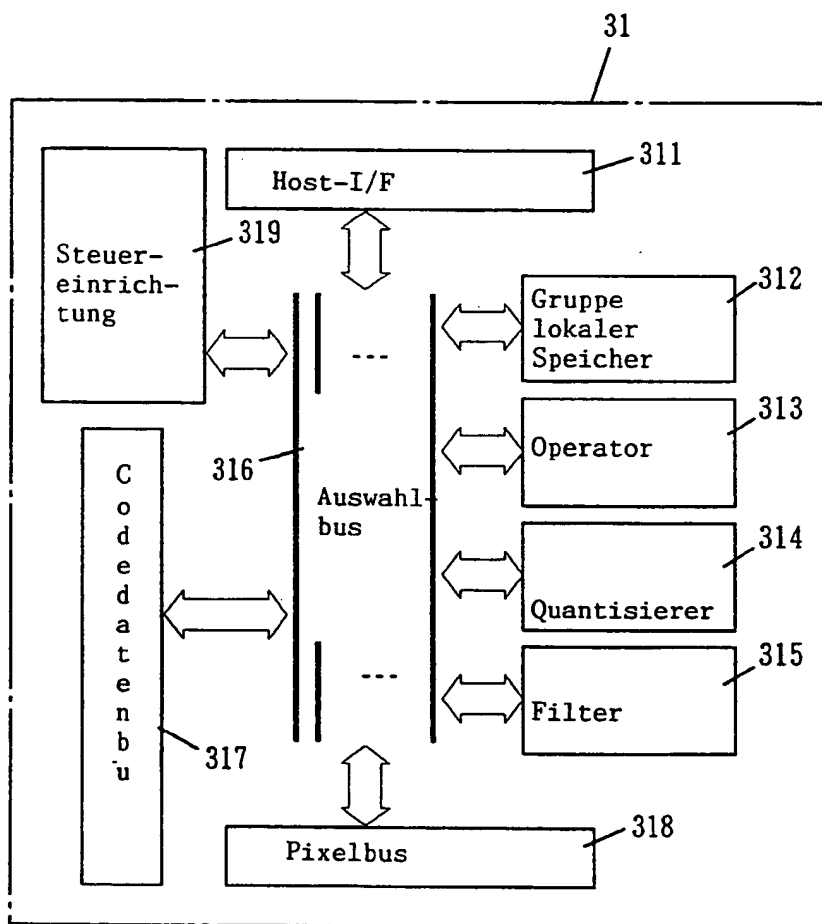


FIG. 19

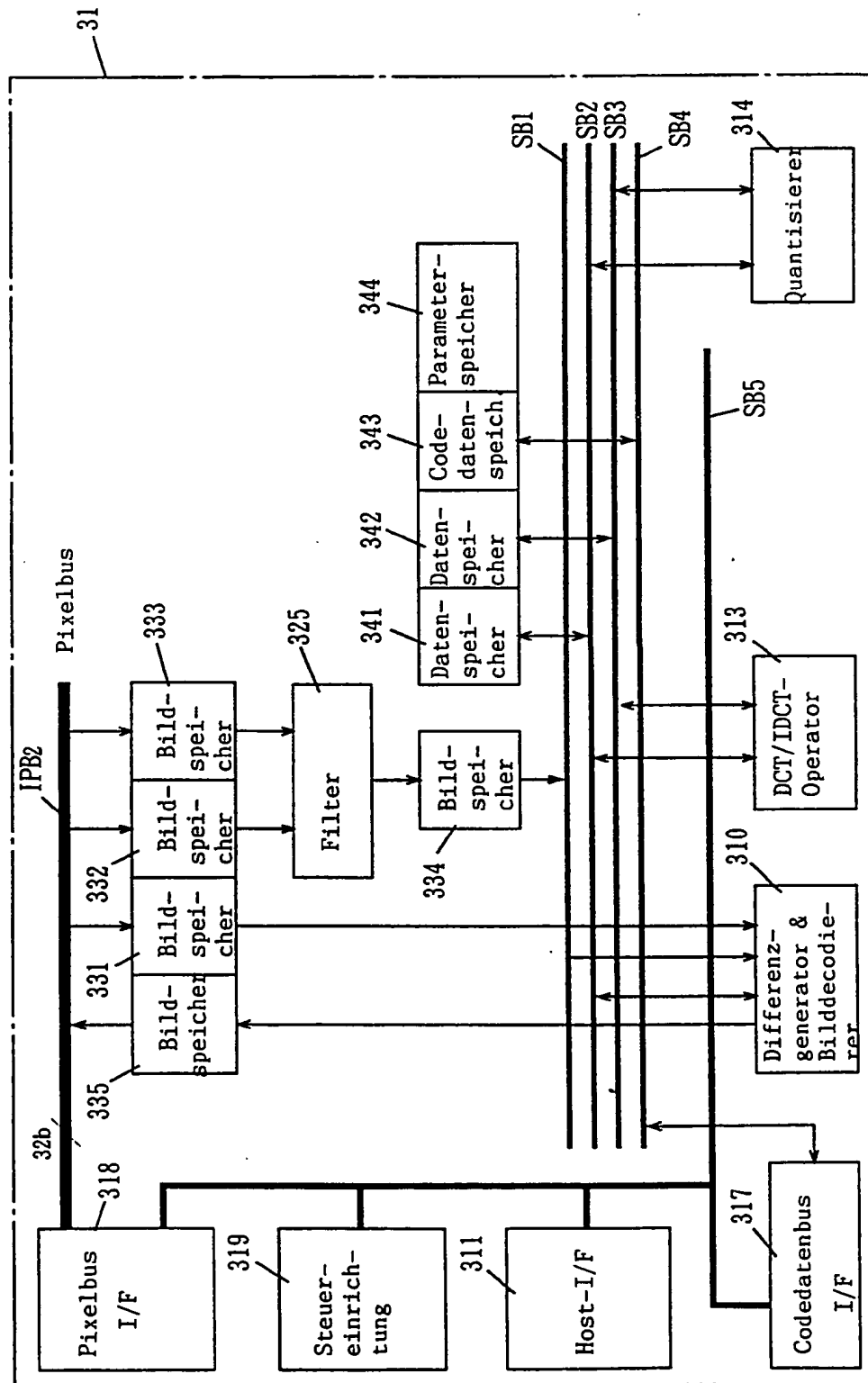


FIG. 20

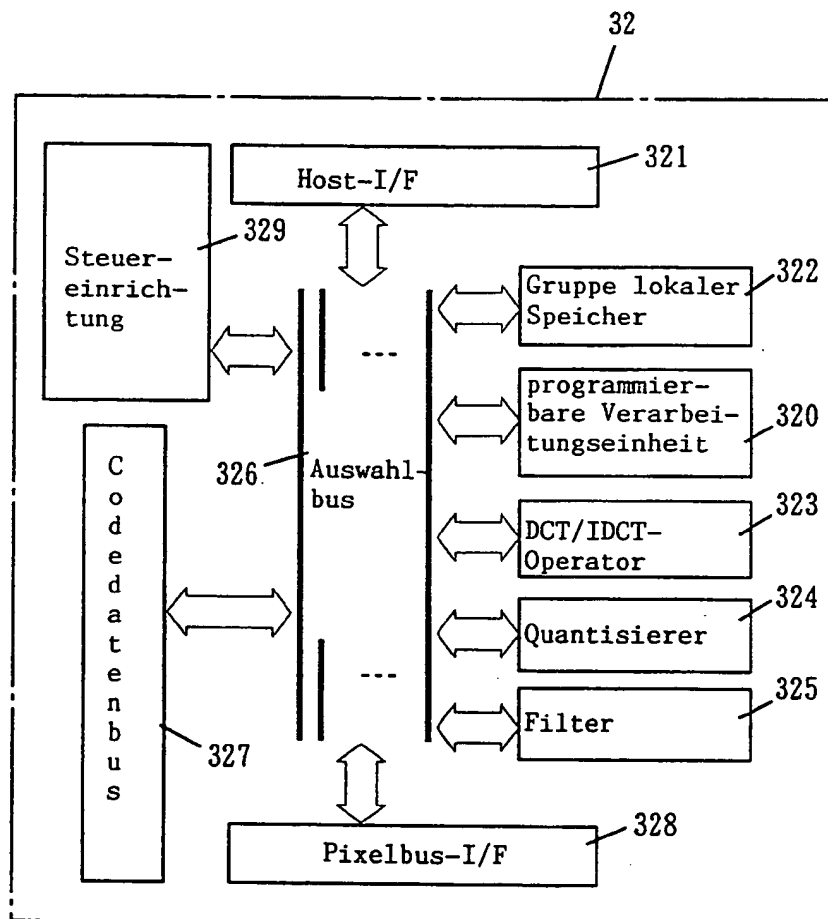


FIG. 21

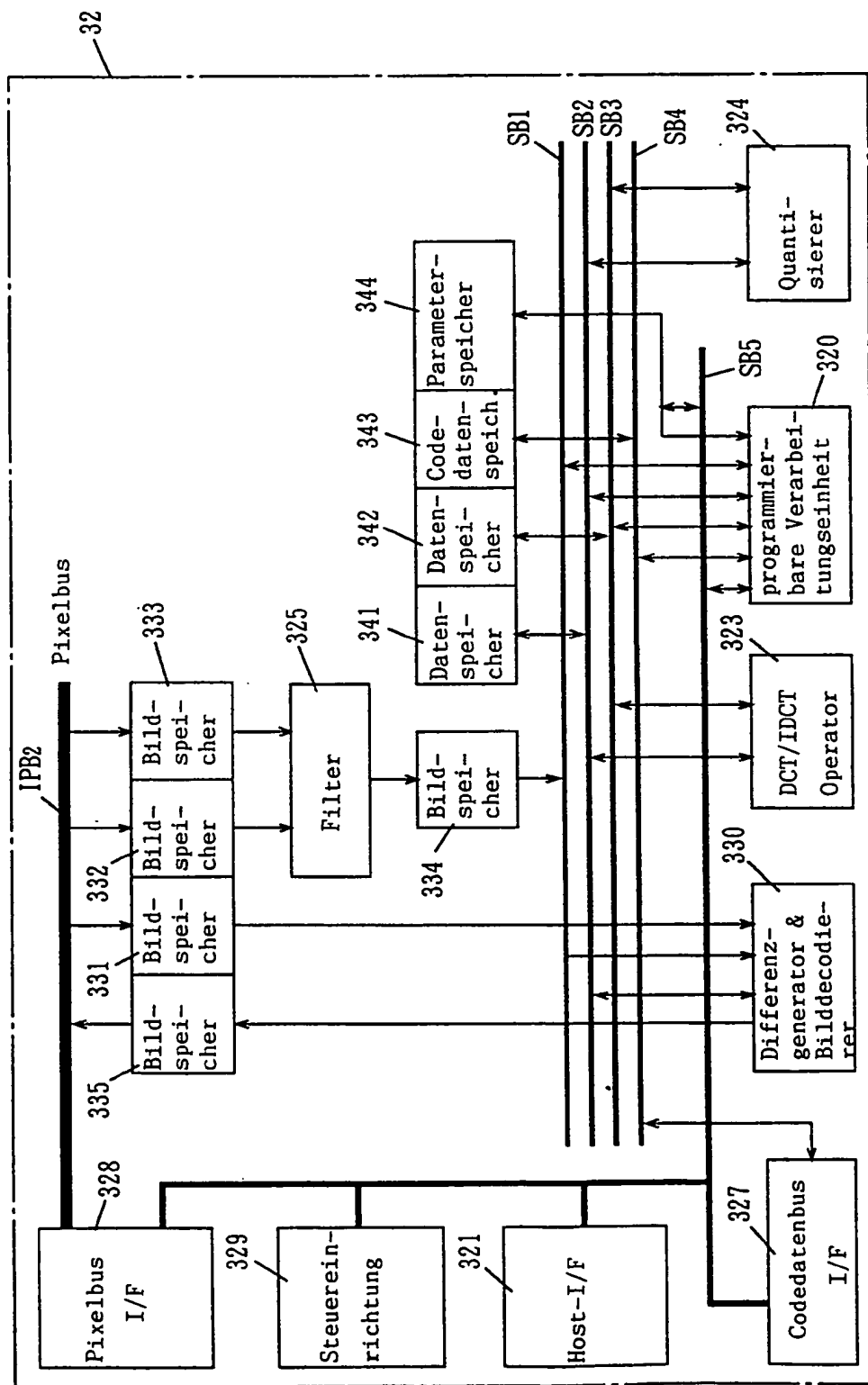


FIG. 22

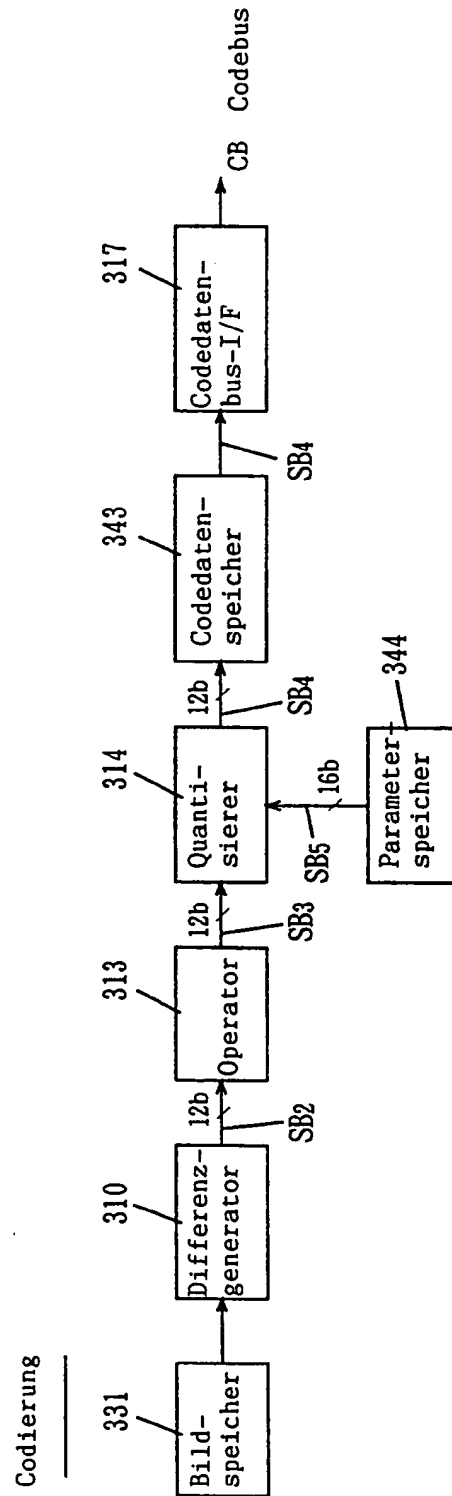




FIG. 23

Codierung

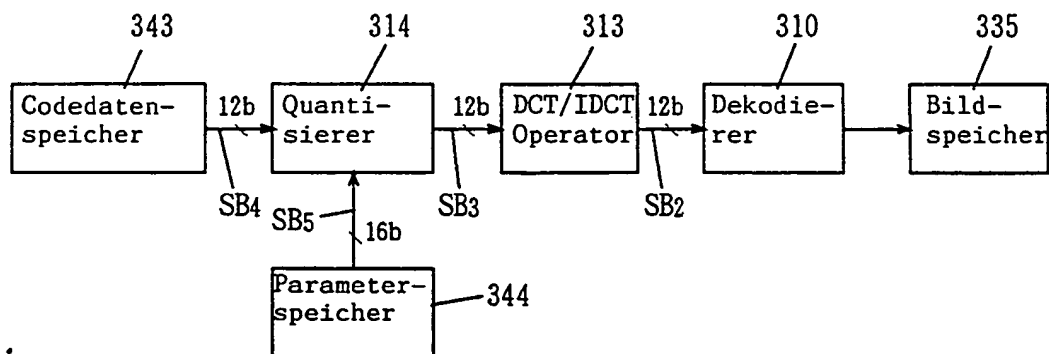


FIG. 24

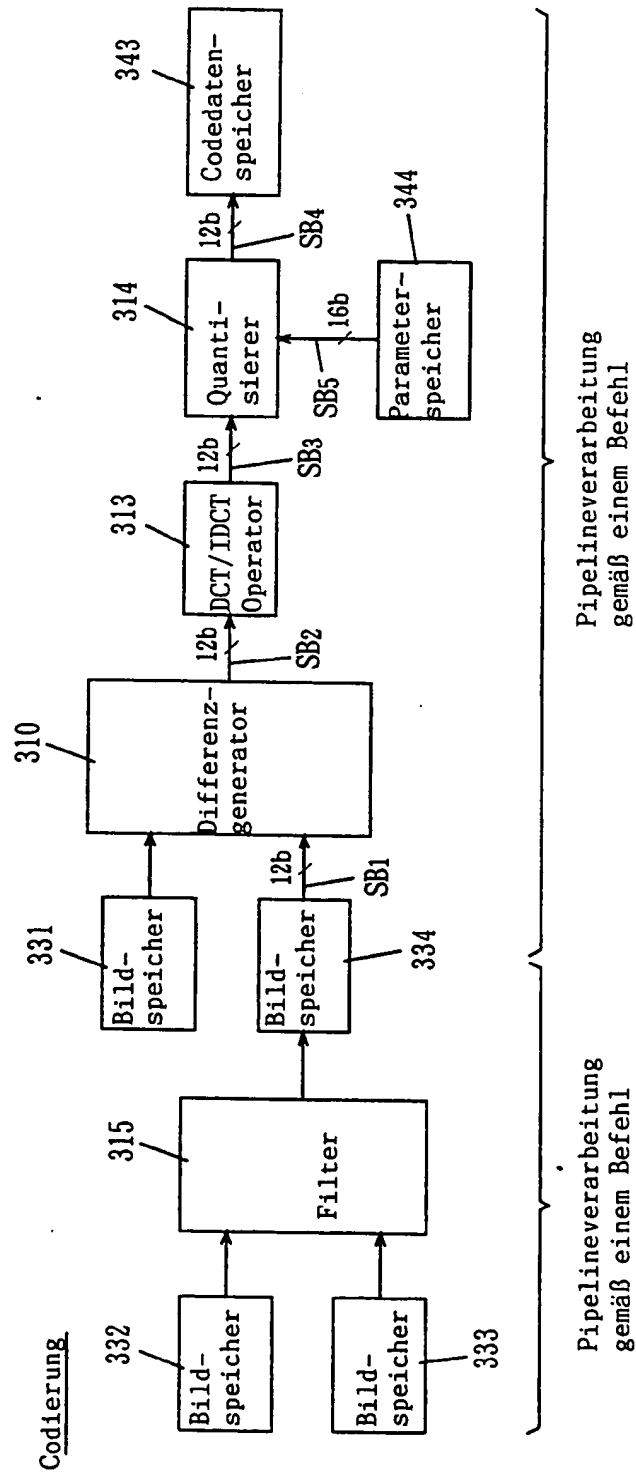


FIG. 25

Codierung

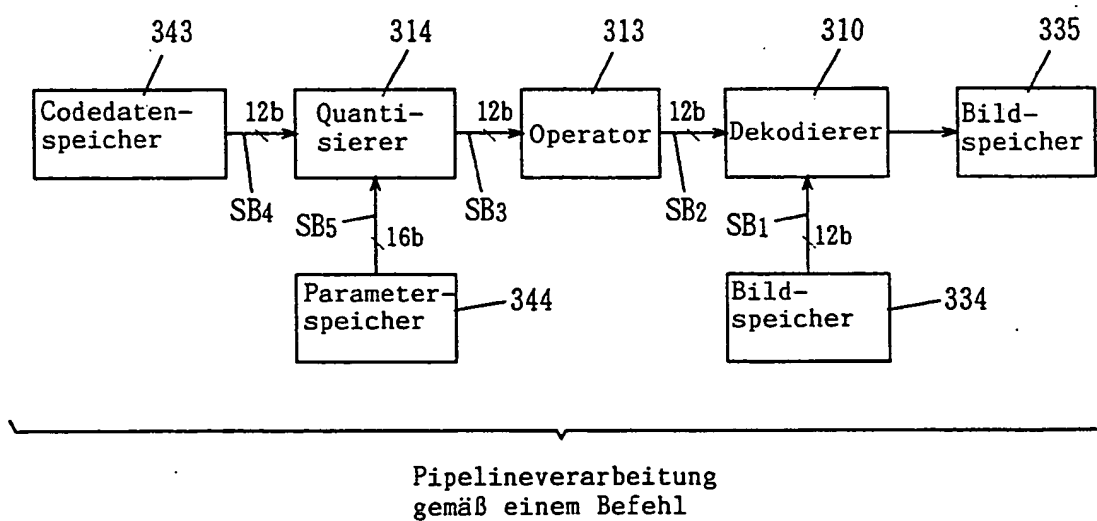


FIG. 26

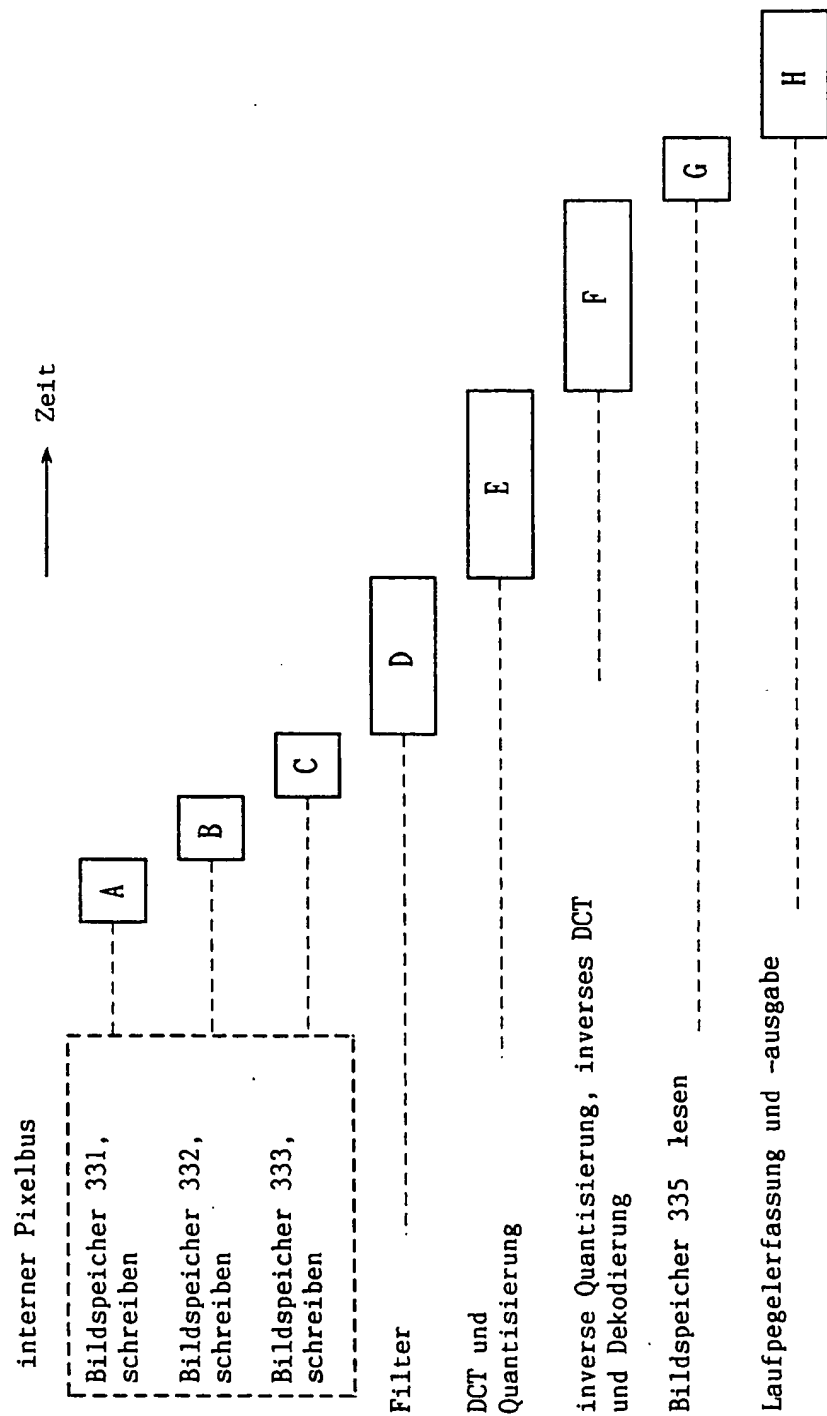


FIG. 27

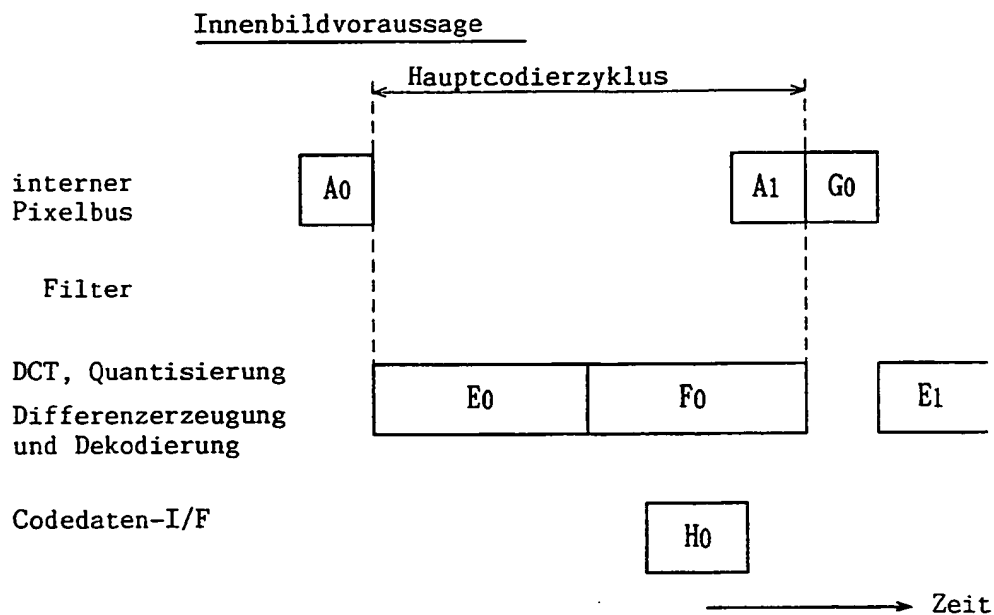


FIG. 28

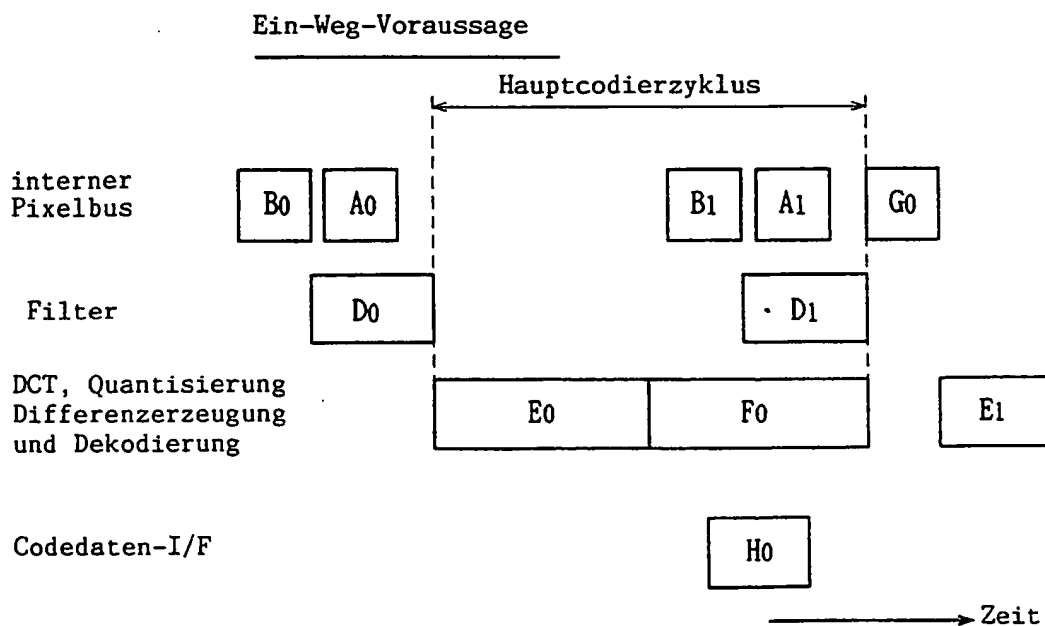


FIG. 29

Zwei-Weg-Voraussage

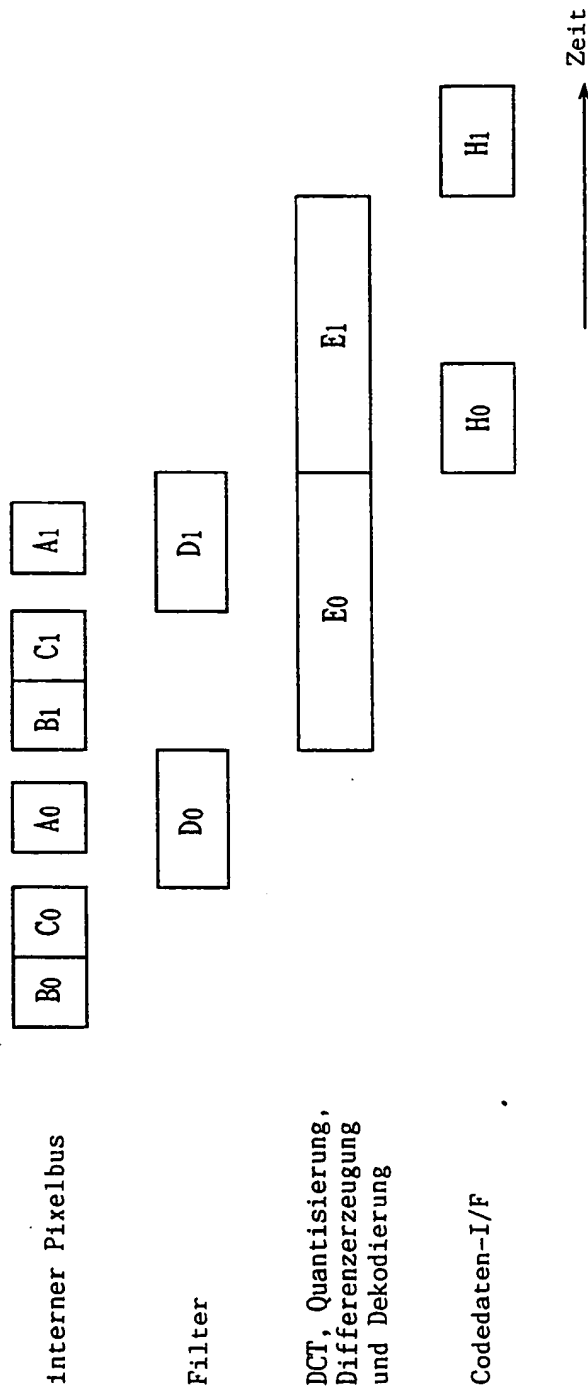


FIG. 30

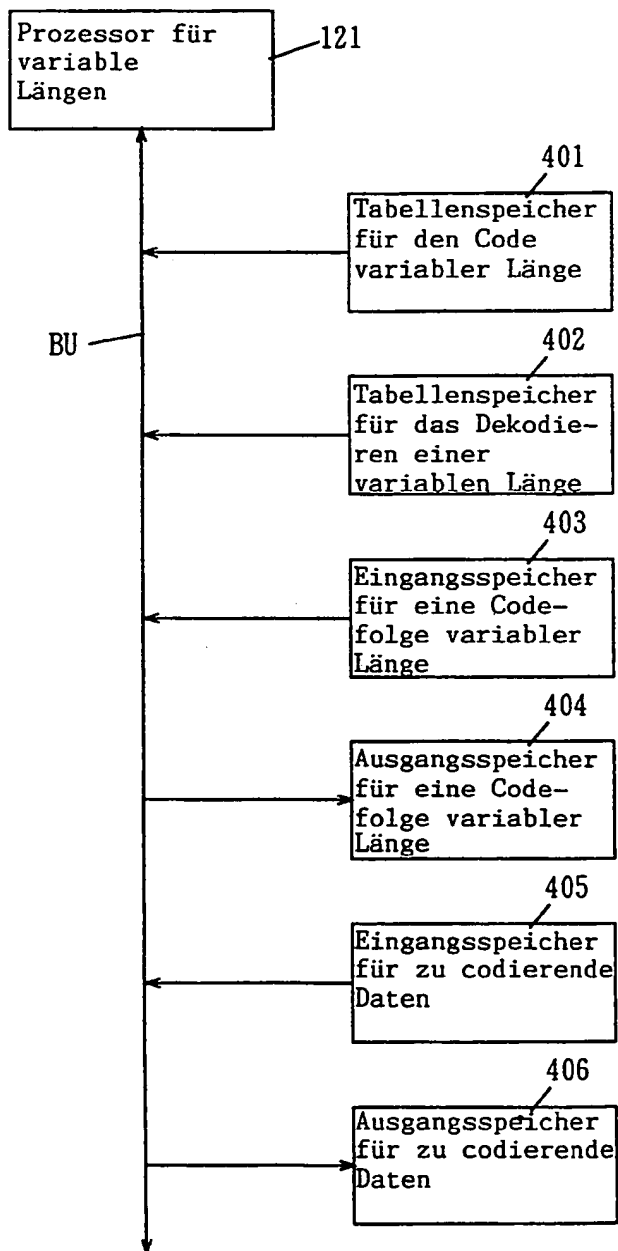


FIG. 31

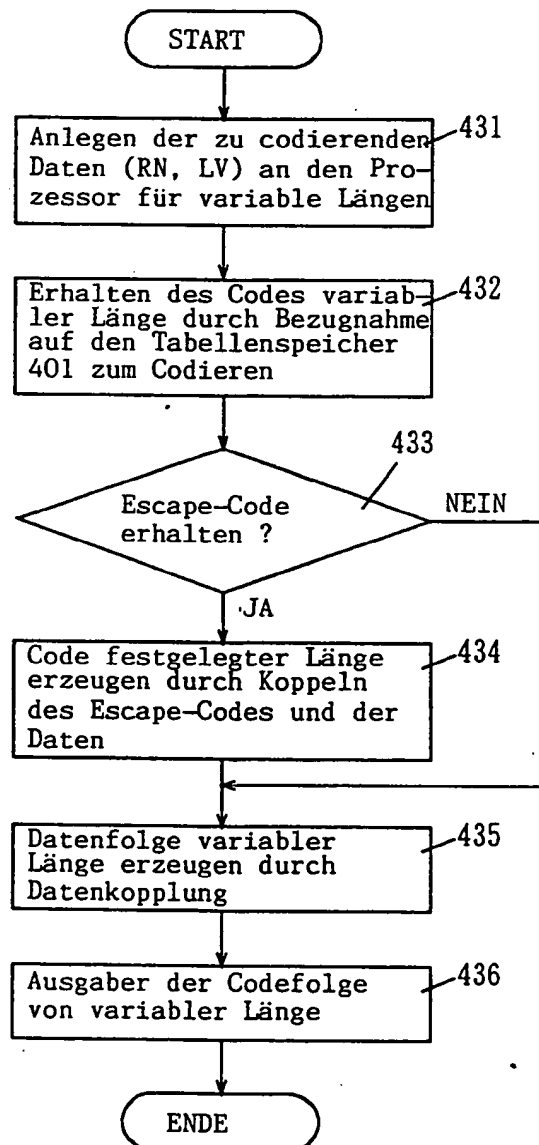




FIG. 32

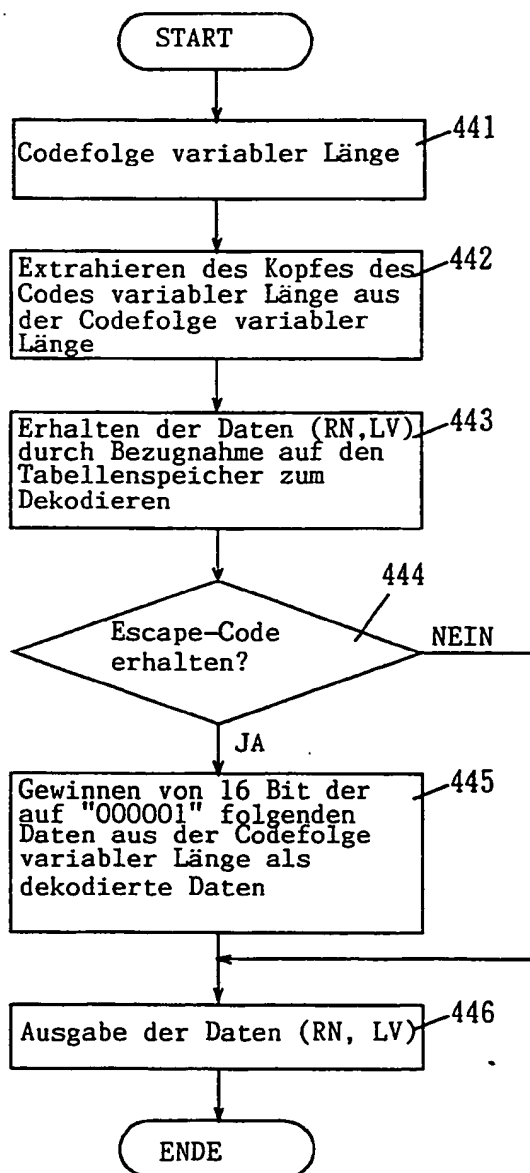


FIG. 33

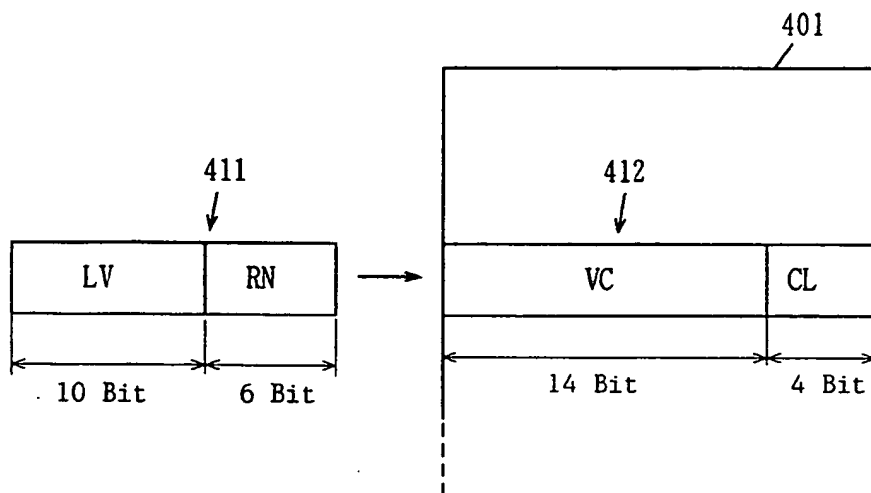


FIG. 34

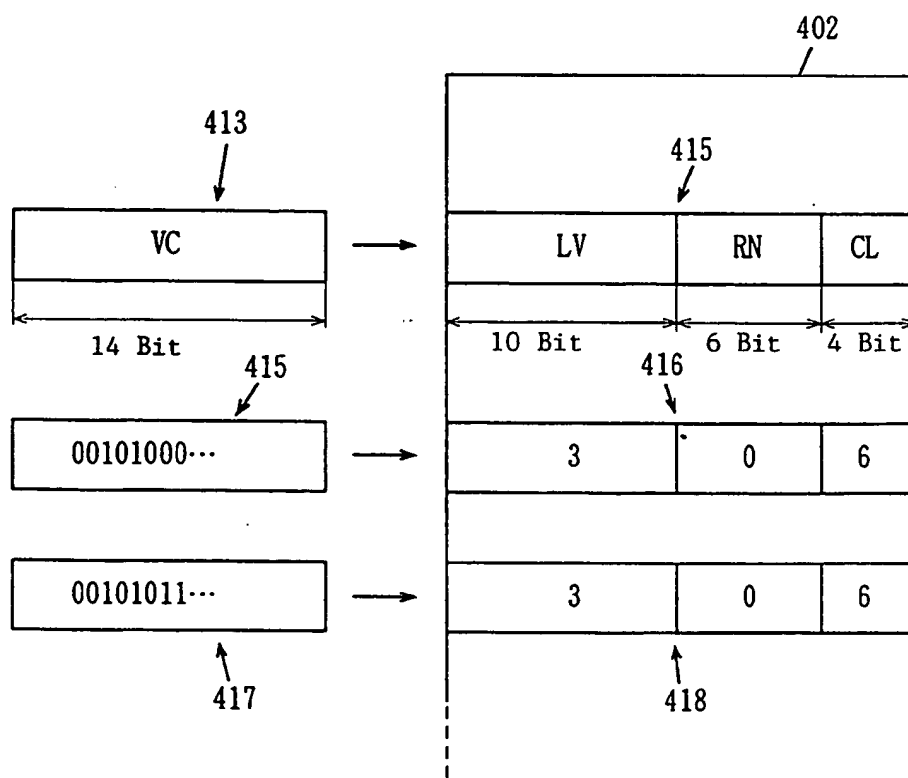


FIG. 35

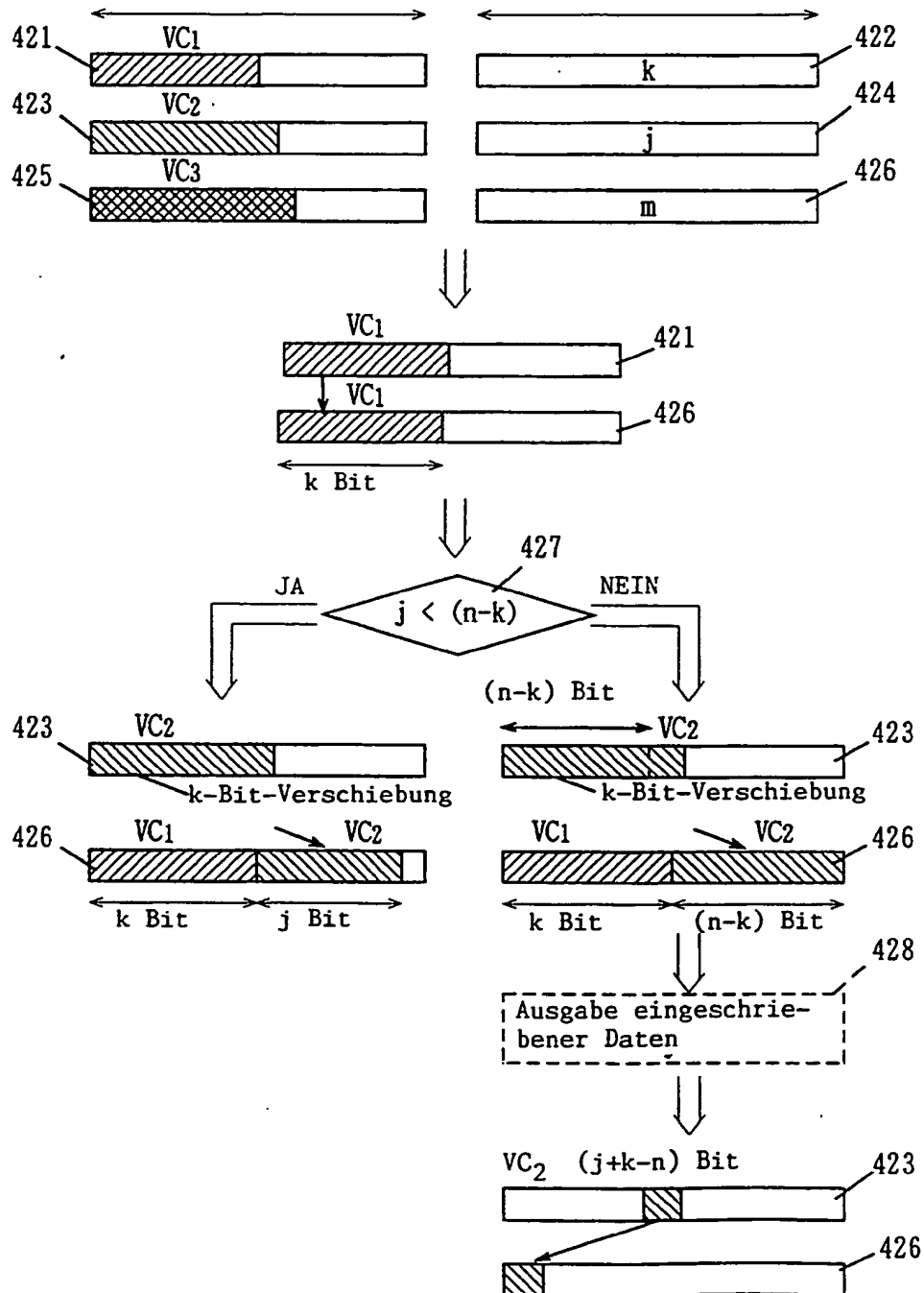


FIG. 36

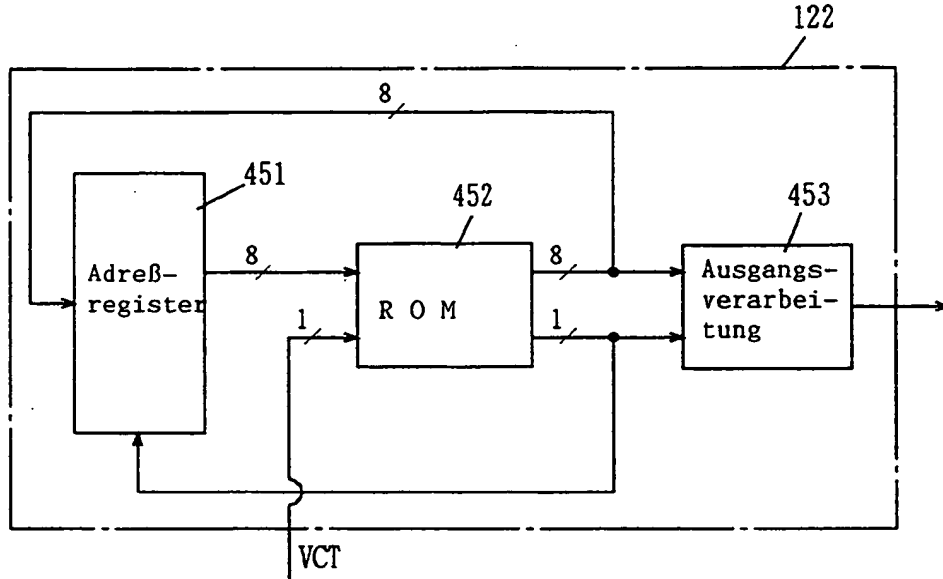


FIG. 37

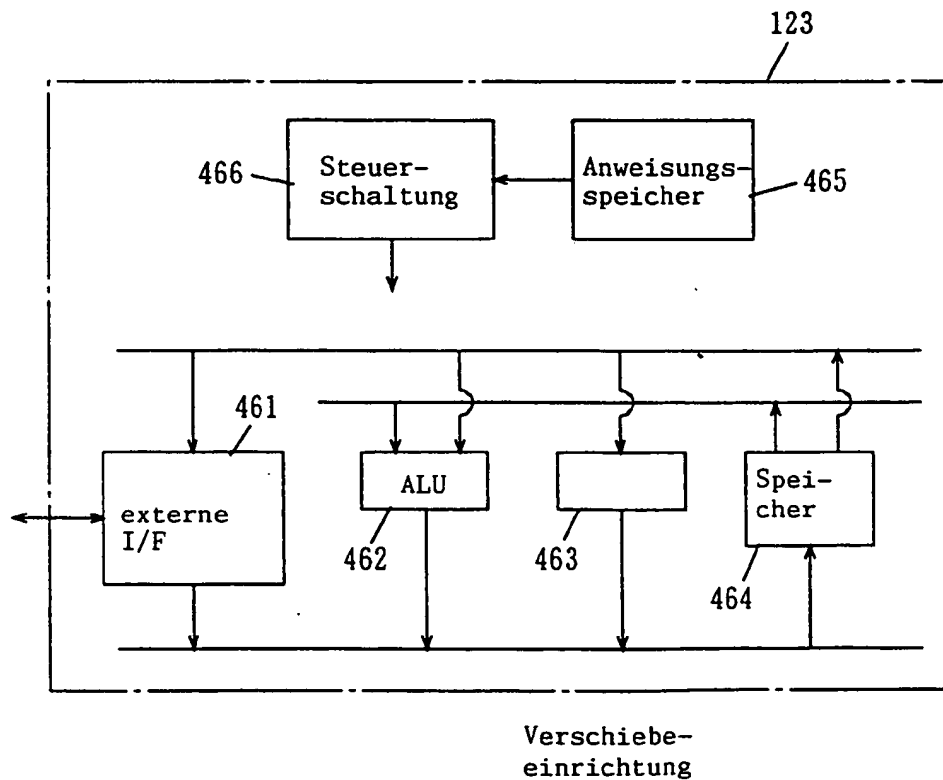


FIG. 38

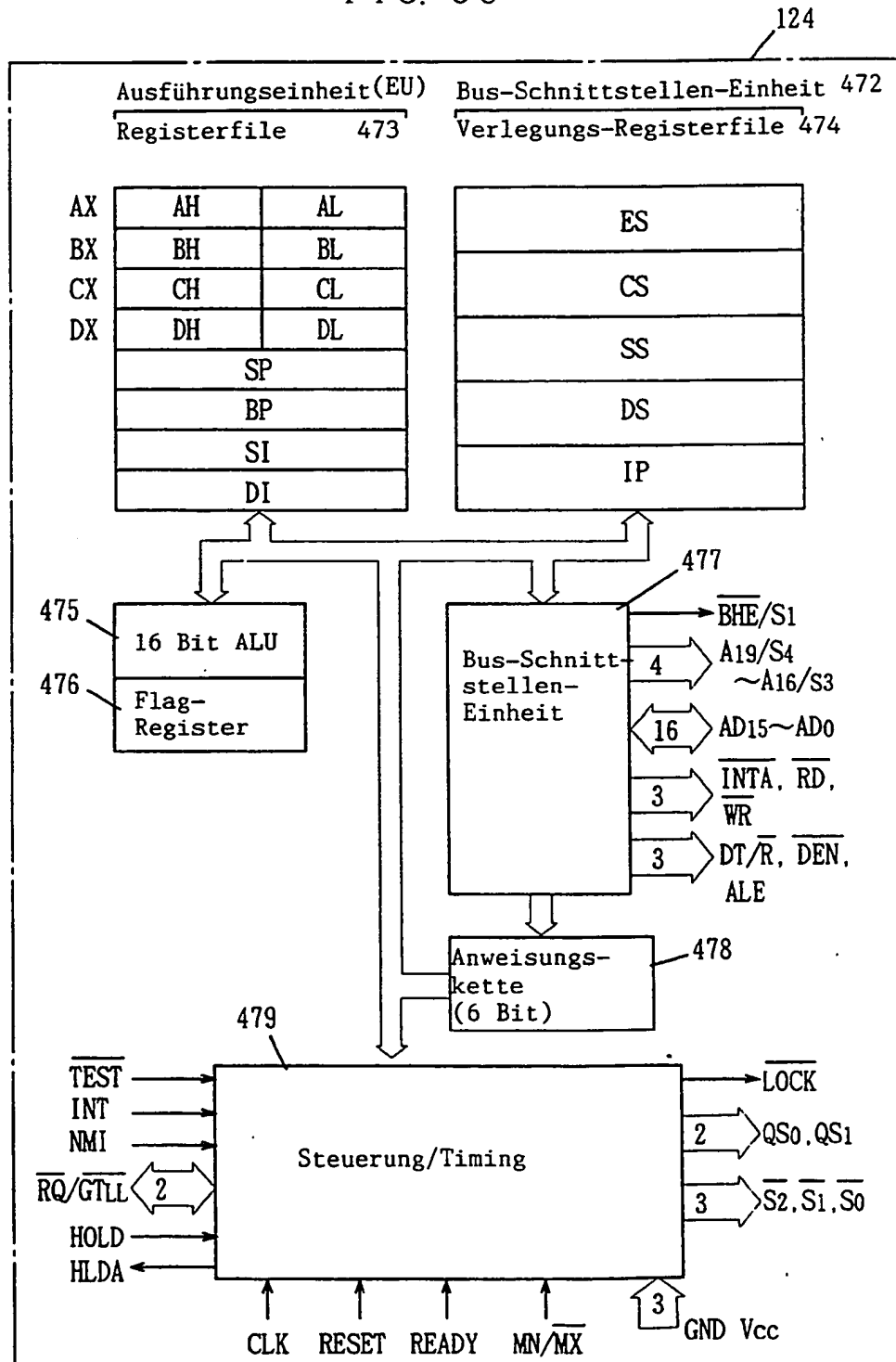


FIG. 39

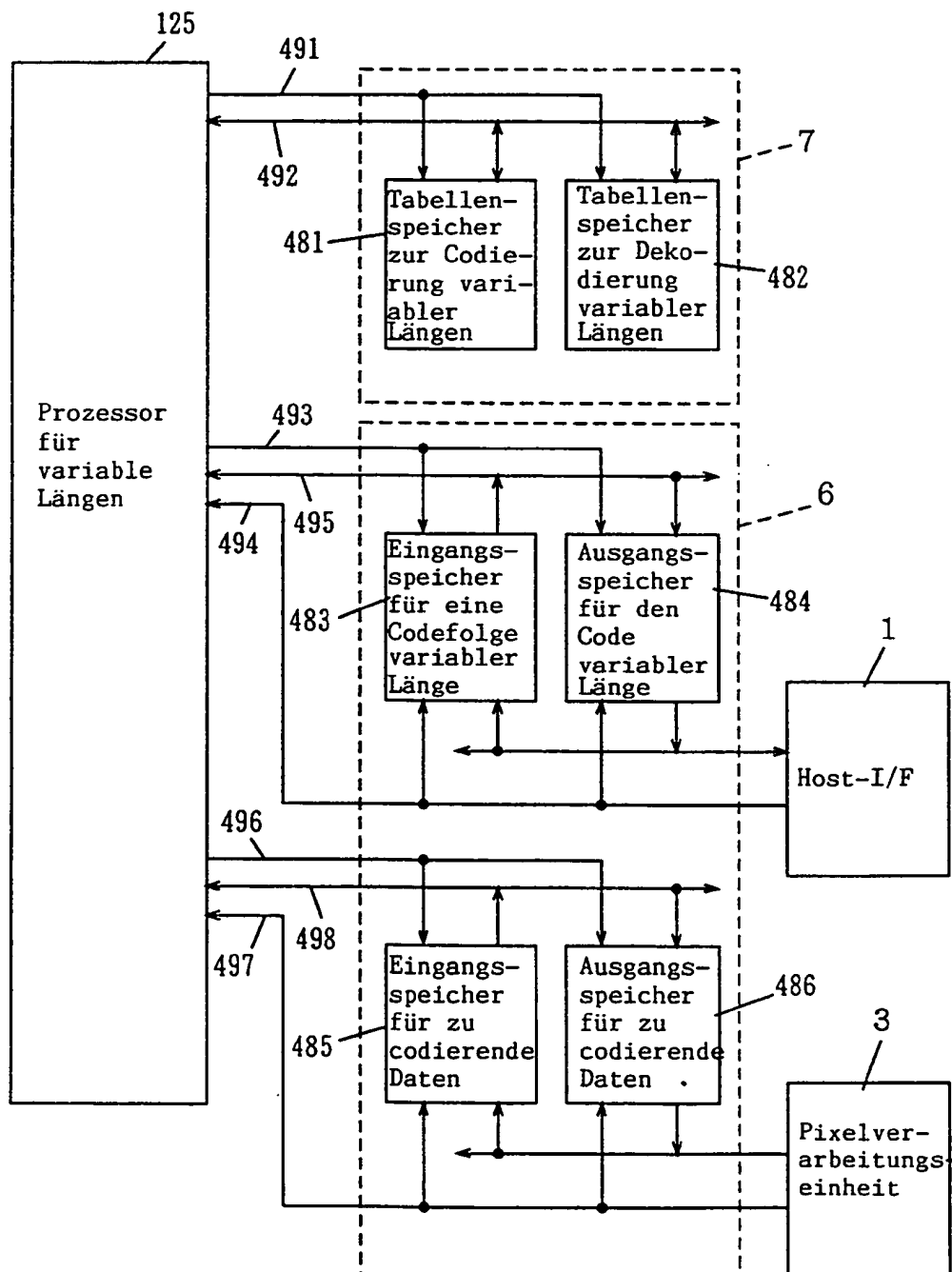


FIG. 40

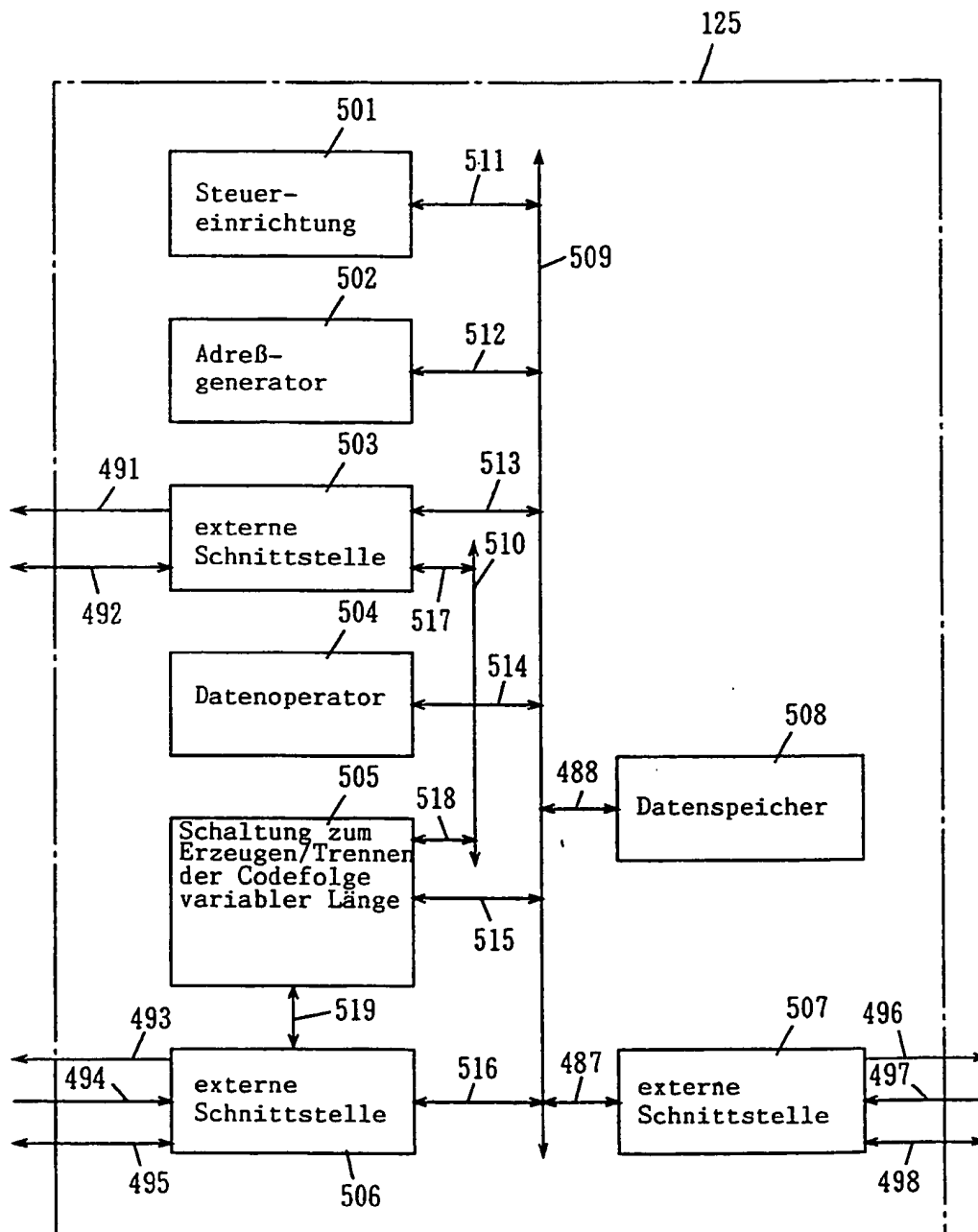


FIG. 41

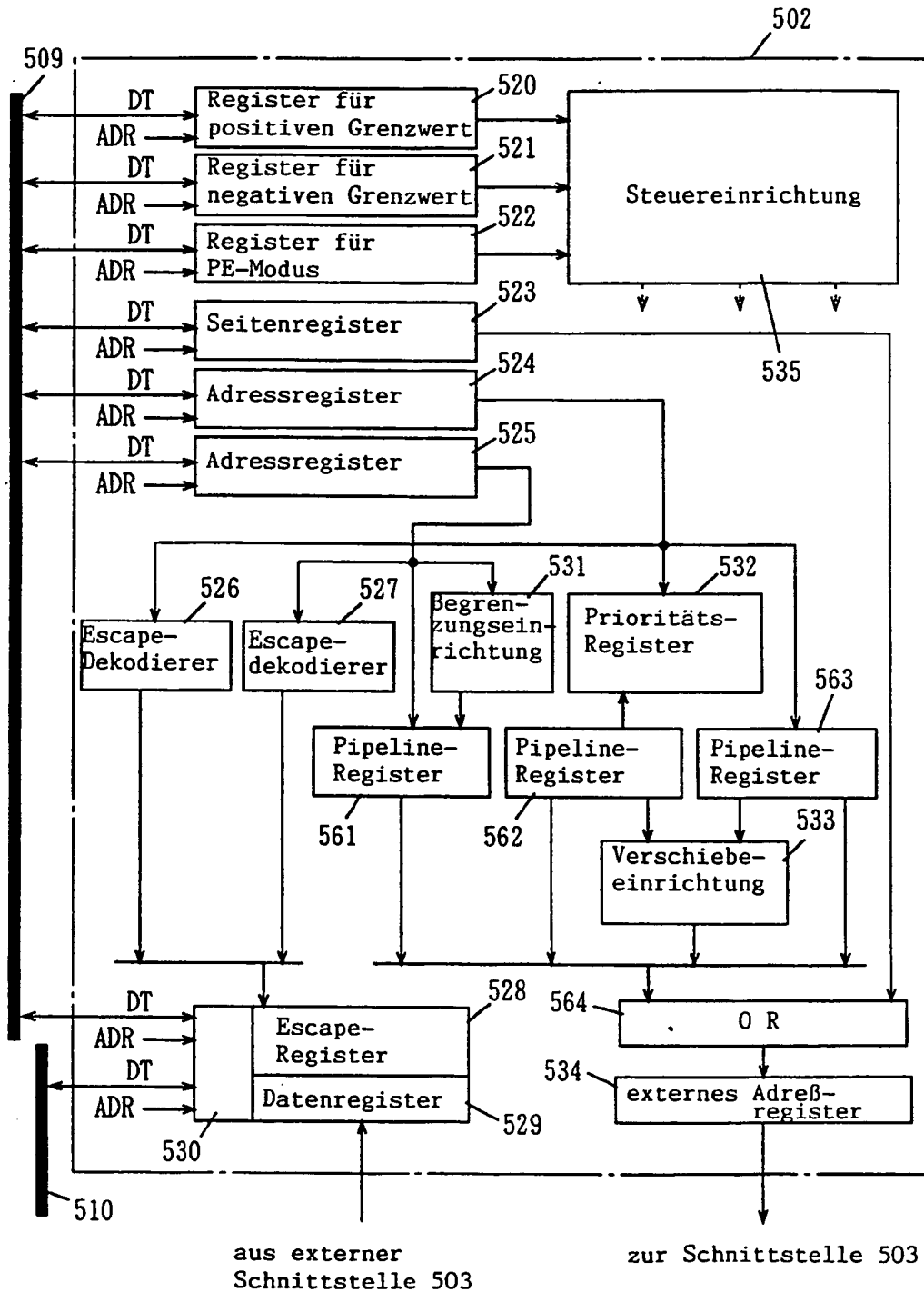




FIG. 42

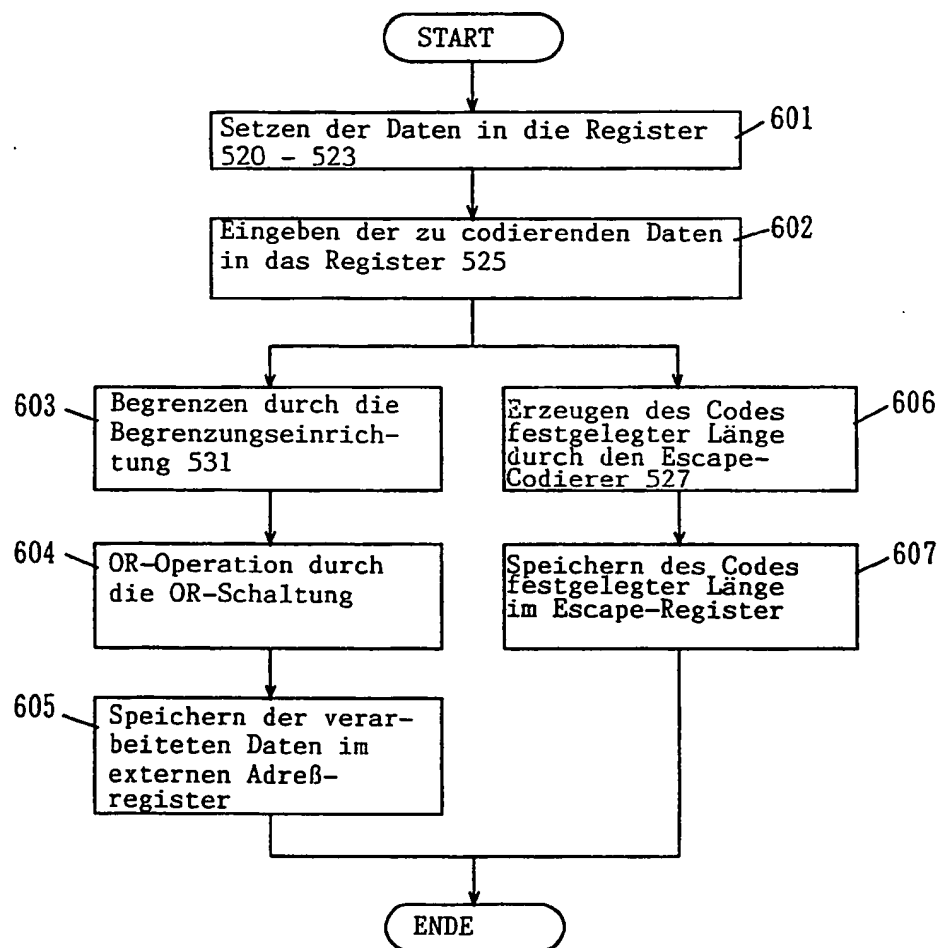


FIG. 43

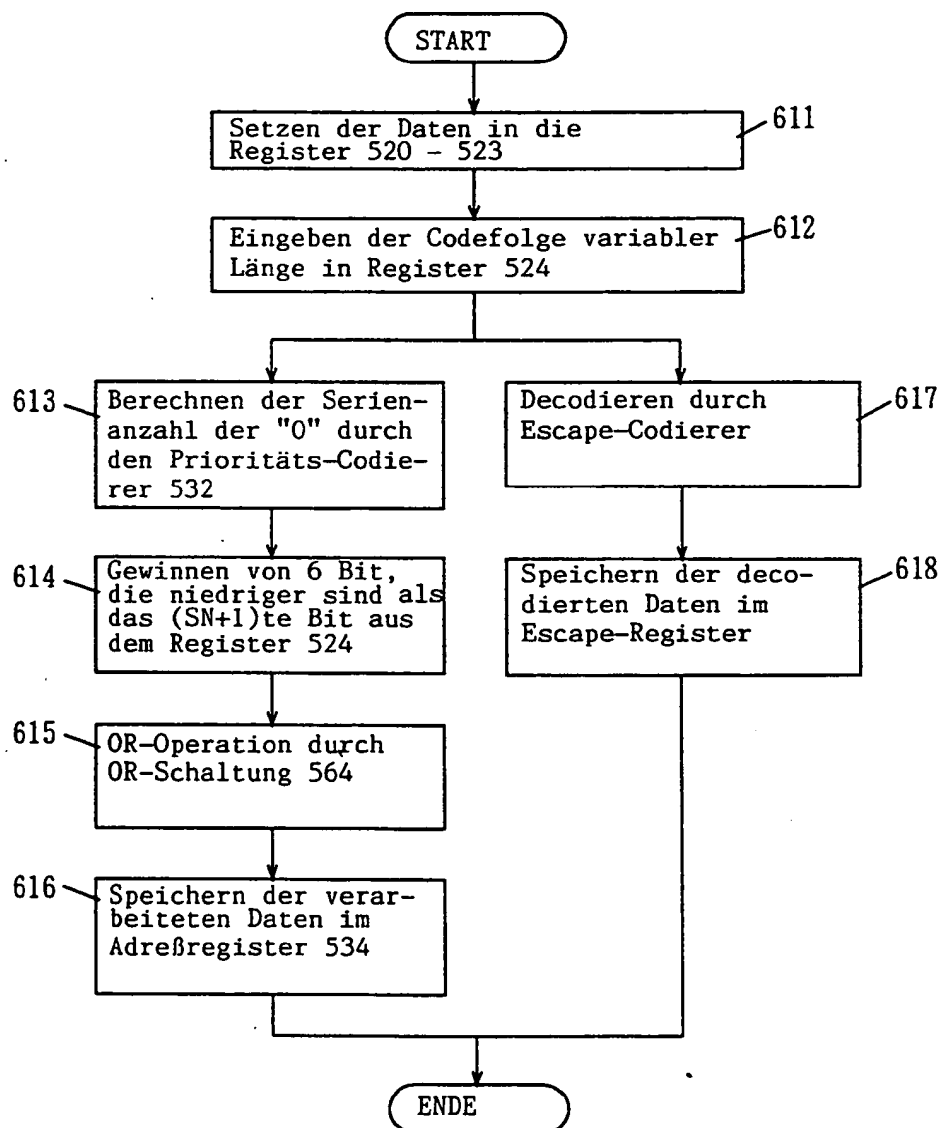


FIG. 44

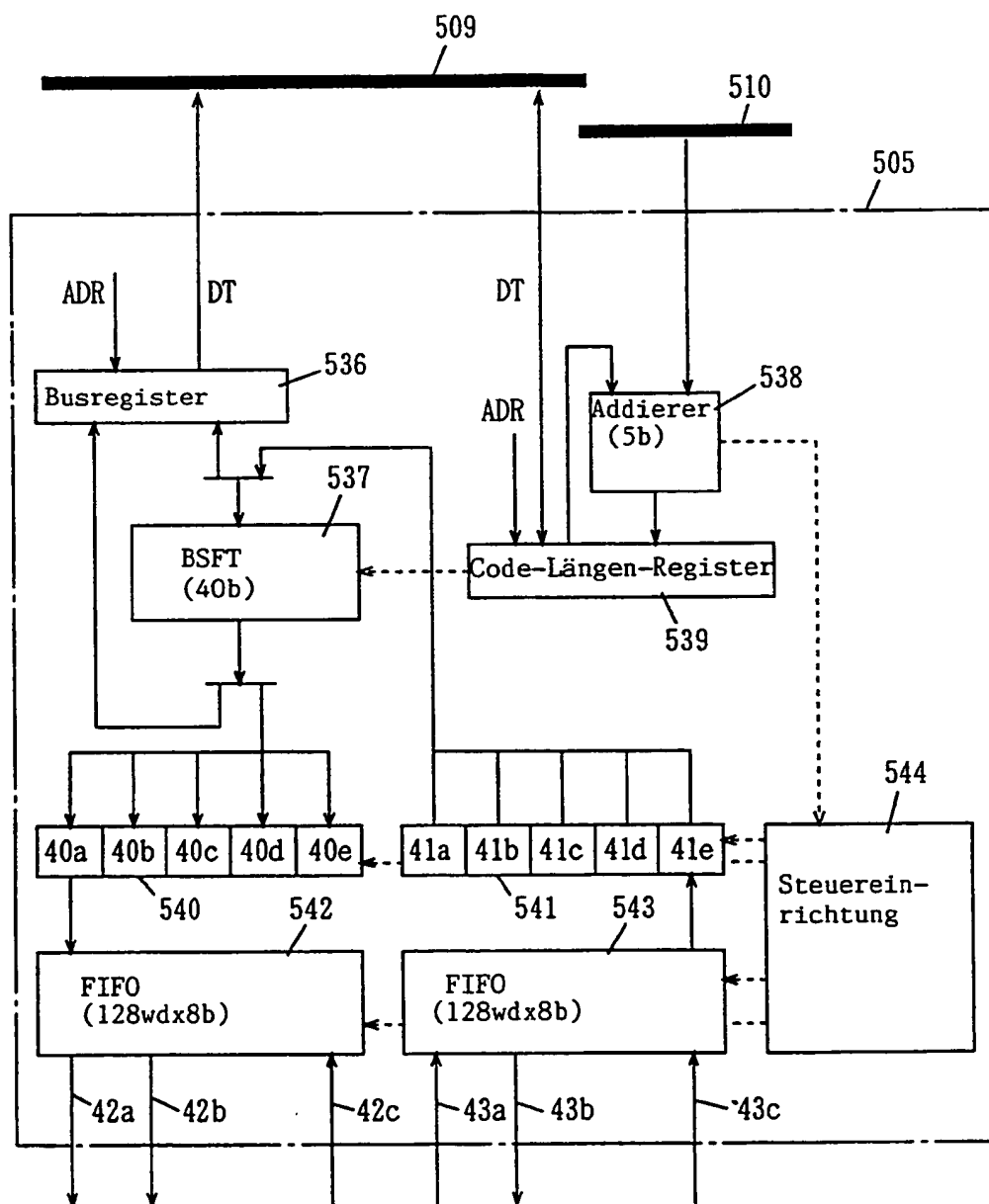


FIG. 45

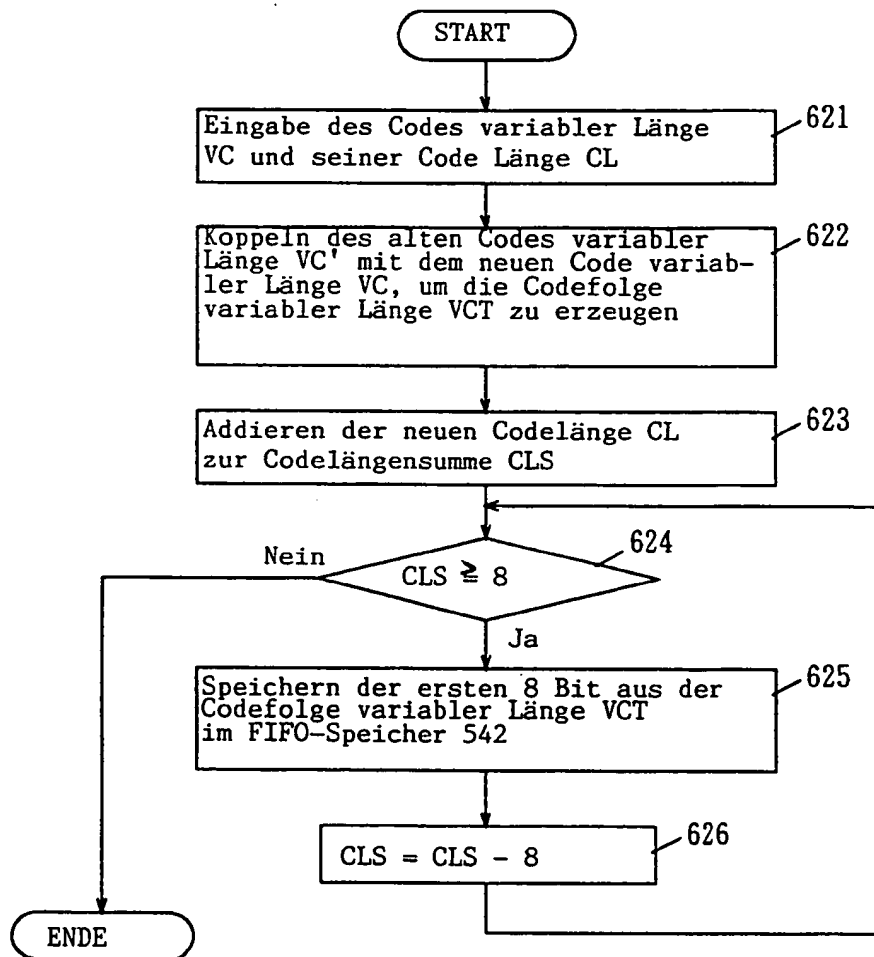


FIG. 46

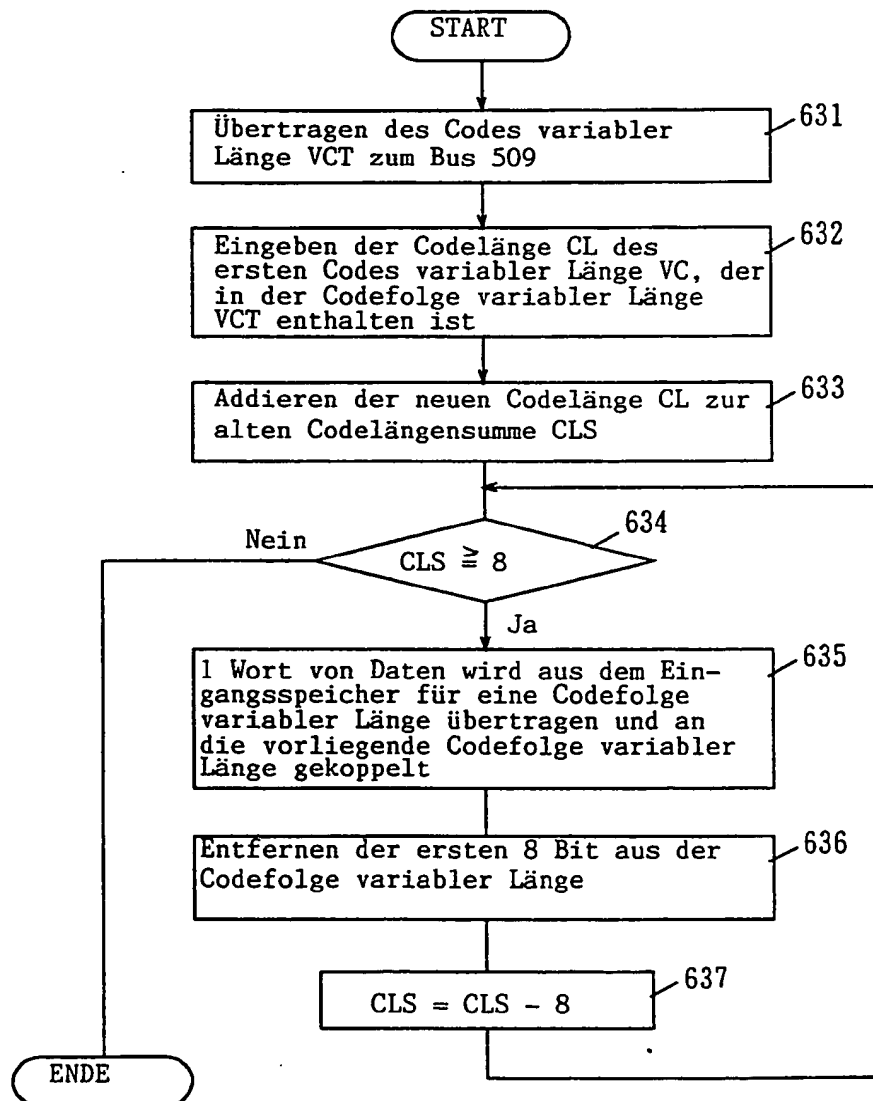


FIG. 47

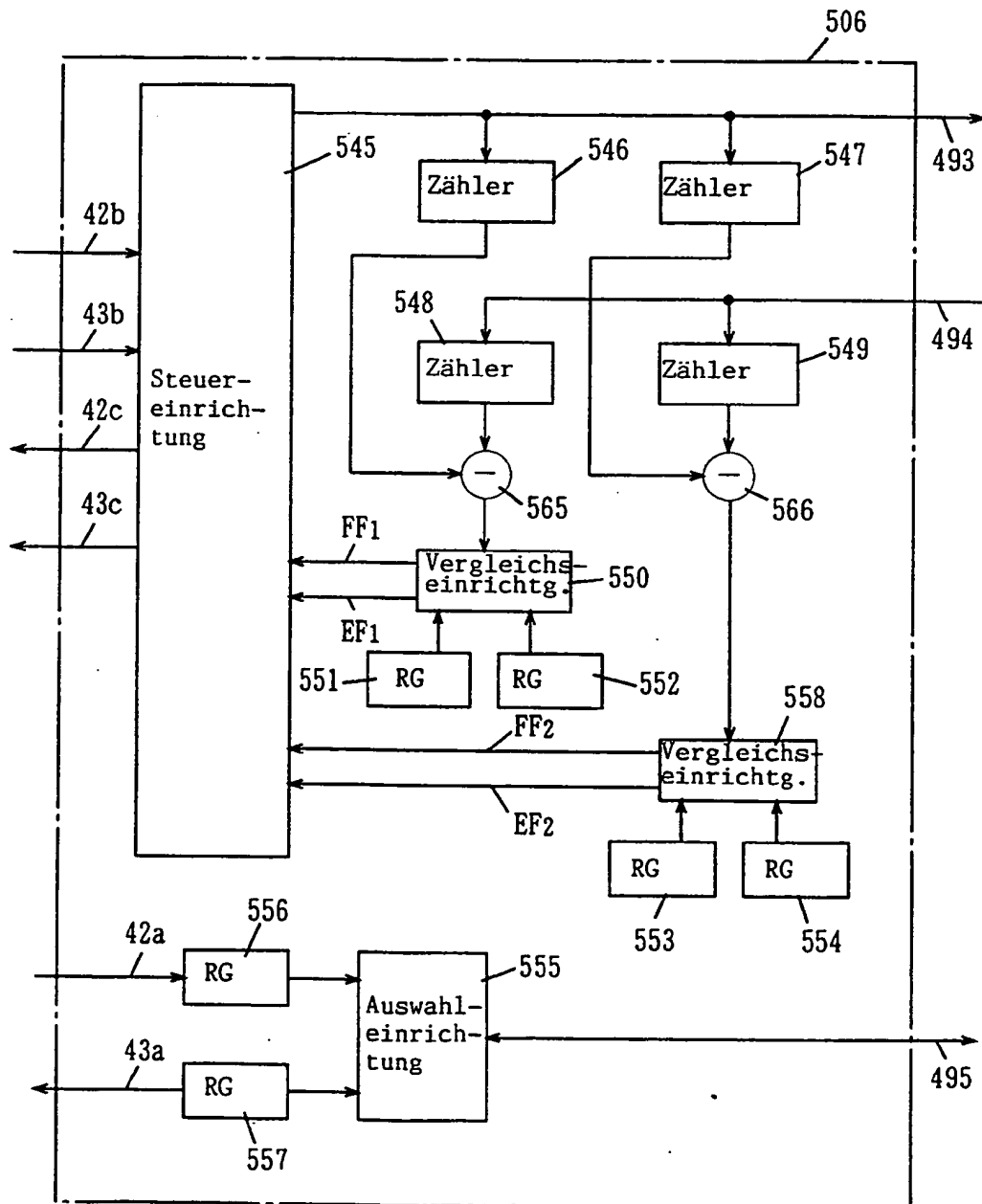


FIG. 48

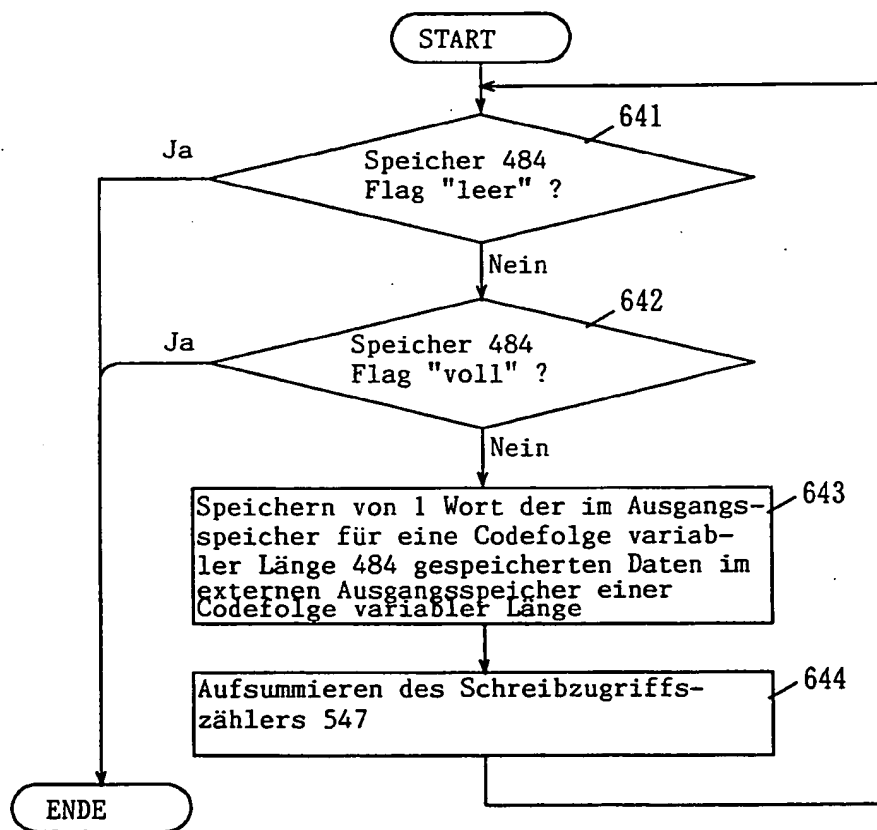


FIG. 49

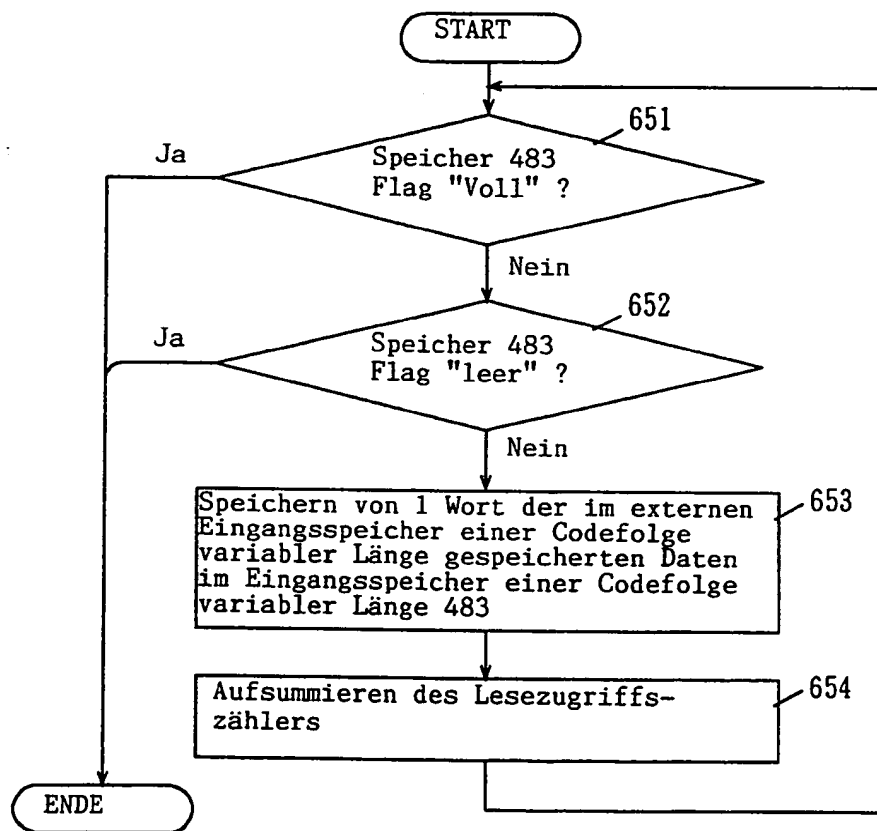




FIG. 50

RN	LV	CL	VC	VC	CL	Gruppe
0	1	2	1s	10	2	GR0
0	1	3	11s	1s	2	
0	2	5	0100s	11s	3	
0	3	6	00101s	0100s	5	GR1
0	4	8	0000110s	0101s	5	
0	5	9	00100110s	011s	4	
0	6	9	00100001s	00100000s	9	GR2
0	7	1 1	0000001010s	00100001s	9	
0	8	1 3	000000011101s	00100010s	9	
0	9	1 3	000000011000s	00100011s	9	
0	1 0	1 3	000000010011s	00100100s	9	
0	1 1	1 3	000000010000s	00100101s	9	
0	1 2	1 4	0000000011010s	00100110s	9	
0	1 3	1 4	0000000011001s	00100111s	9	
0	1 4	1 4	0000000011000s	00101s	6	
0	1 5	1 4	0000000010111s	00110s	6	
1	1	4	011s	00111s	6	GR3
1	2	7	000110s	000100s	7	
1	3	9	00100101s	000101s	7	
1	4	1 1	0000001100s	000110s	7	
1	5	1 3	000000011011s	000111s	7	GR4
1	6	1 4	0000000010110s	0000100s	8	
1	7	1 4	0000000010101s	0000101s	8	
2	1	5	0101s	0000110s	8	
2	2	8	0000100s	0000111s	8	GR5
2	3	1 1	0000001011s	000001	6	
2	4	1 3	000000010100s	0000001000s	1 1	
2	5	1 4	0000000010100s	0000001001s	1 1	
3	1	6	00111s	0000001010s	1 1	GR6
3	2	9	00100100s	0000001011s	1 1	
3	3	1 3	000000011100s	0000001100s	1 1	
3	4	1 4	0000000010011s	0000001101s	1 1	
4	1	6	00110s	0000001110s	1 1	GR7
4	2	1 1	0000001111s	0000001111s	1 1	
4	3	1 3	000000010010s	000000010000s	1 3	
5	1	7	000111s	000000010001s	1 3	
5	2	1 1	0000001001s	000000010010s	1 3	
5	3	1 4	0000000010010s	000000010011s	1 3	
6	1	7	000101s	000000010100s	1 3	
6	2	1 3	000000010101s	000000010101s	1 3	
7	1	7	000100s	000000010110s	1 3	
7	2	1 3	000000011110s	000000010111s	1 3	
8	1	8	0000111s	000000011000s	1 3	GR8
8	2	1 3	000000010001s	000000011001s	1 3	
9	1	8	0000101s	000000011010s	1 3	
9	2	1 4	0000000010001s	000000011011s	1 3	
1 0	1	9	00100111s	000000011100s	1 3	
1 0	2	1 4	0000000010000s	000000011101s	1 3	
1 1	1	9	00100011s	000000011110s	1 3	
1 2	1	9	00100010s	000000011111s	1 3	
1 3	1	9	00100000s	0000000010000s	1 4	
1 4	1	1 1	0000001110s	0000000010001s	1 4	
1 5	1	1 1	0000001101s	0000000010010s	1 4	GR8
1 6	1	1 1	0000001000s	0000000010011s	1 4	
1 7	1	1 3	000000011111s	0000000010100s	1 4	
1 8	1	1 3	000000011010s	0000000010101s	1 4	
1 9	1	1 3	000000011001s	0000000010110s	1 4	
2 0	1	1 3	000000010111s	0000000010111s	1 4	
2 1	1	1 3	000000010110s	0000000011000s	1 4	
2 2	1	1 4	0000000011111s	0000000011001s	1 4	
2 3	1	1 4	0000000011110s	0000000011010s	1 4	
2 4	1	1 4	0000000011101s	0000000011011s	1 4	
2 5	1	1 4	0000000011100s	0000000011100s	1 4	GR8
2 6	1	1 4	0000000011011s	0000000011101s	1 4	
		6	000001	0000000011110s	1 4	
		2	10	0000000011111s	1 4	

FIG. 51

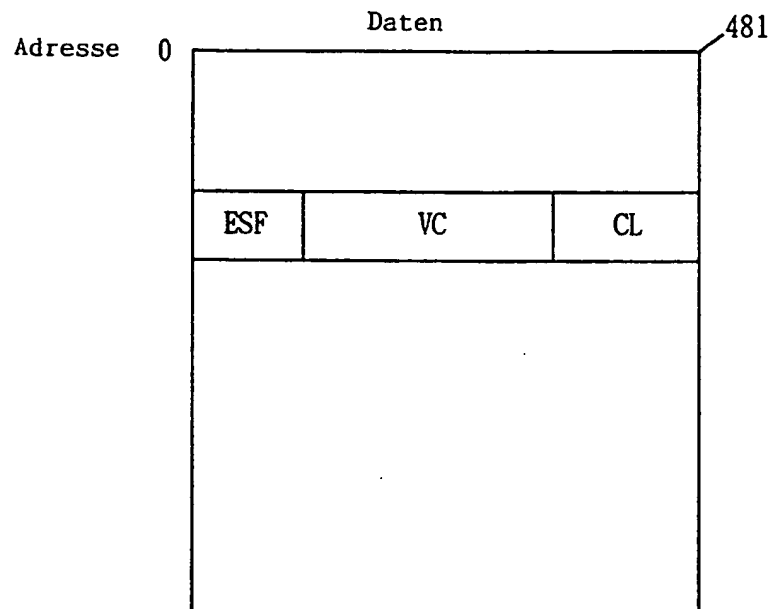


FIG. 52

zu codierende Daten

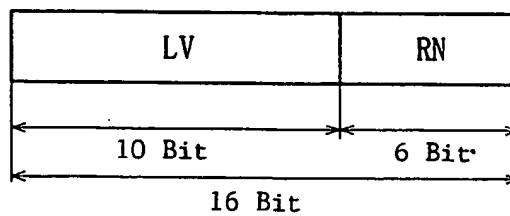


FIG. 53

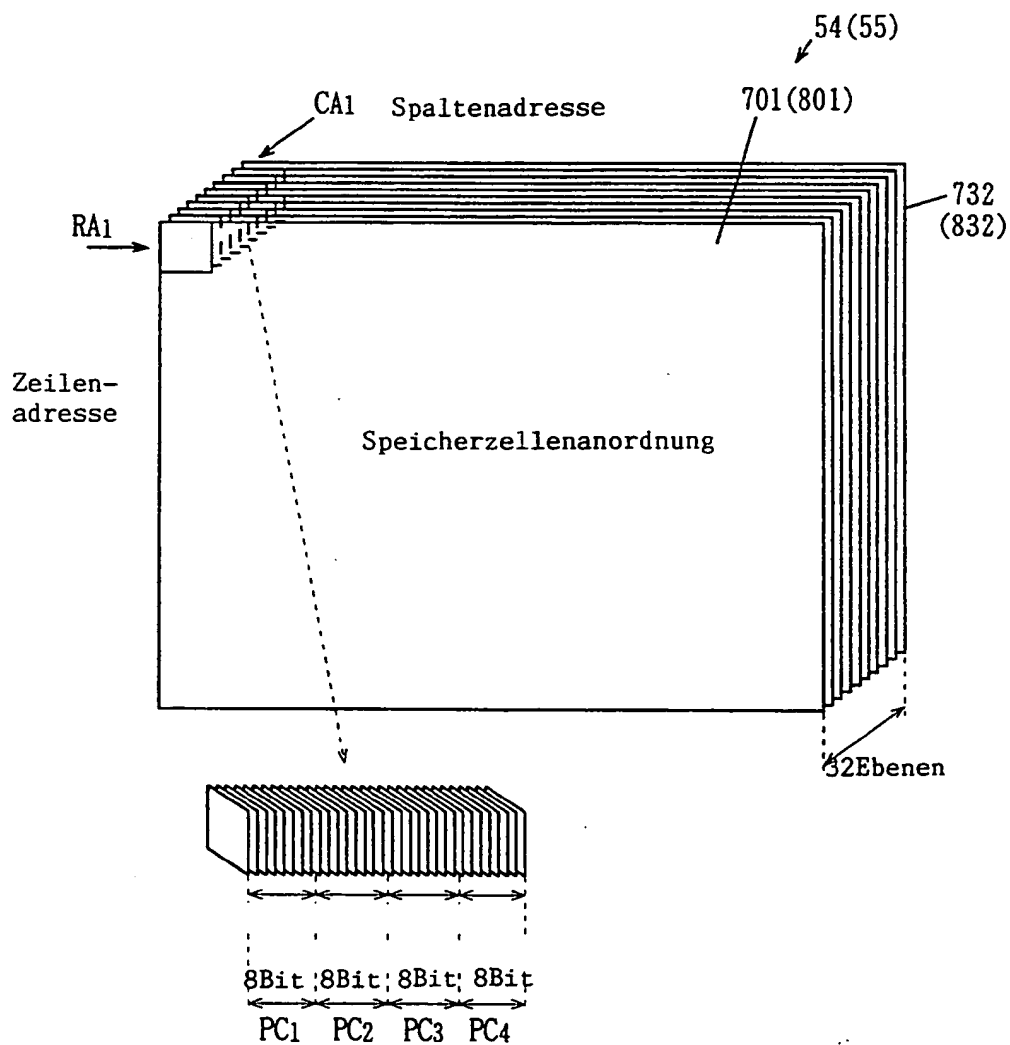
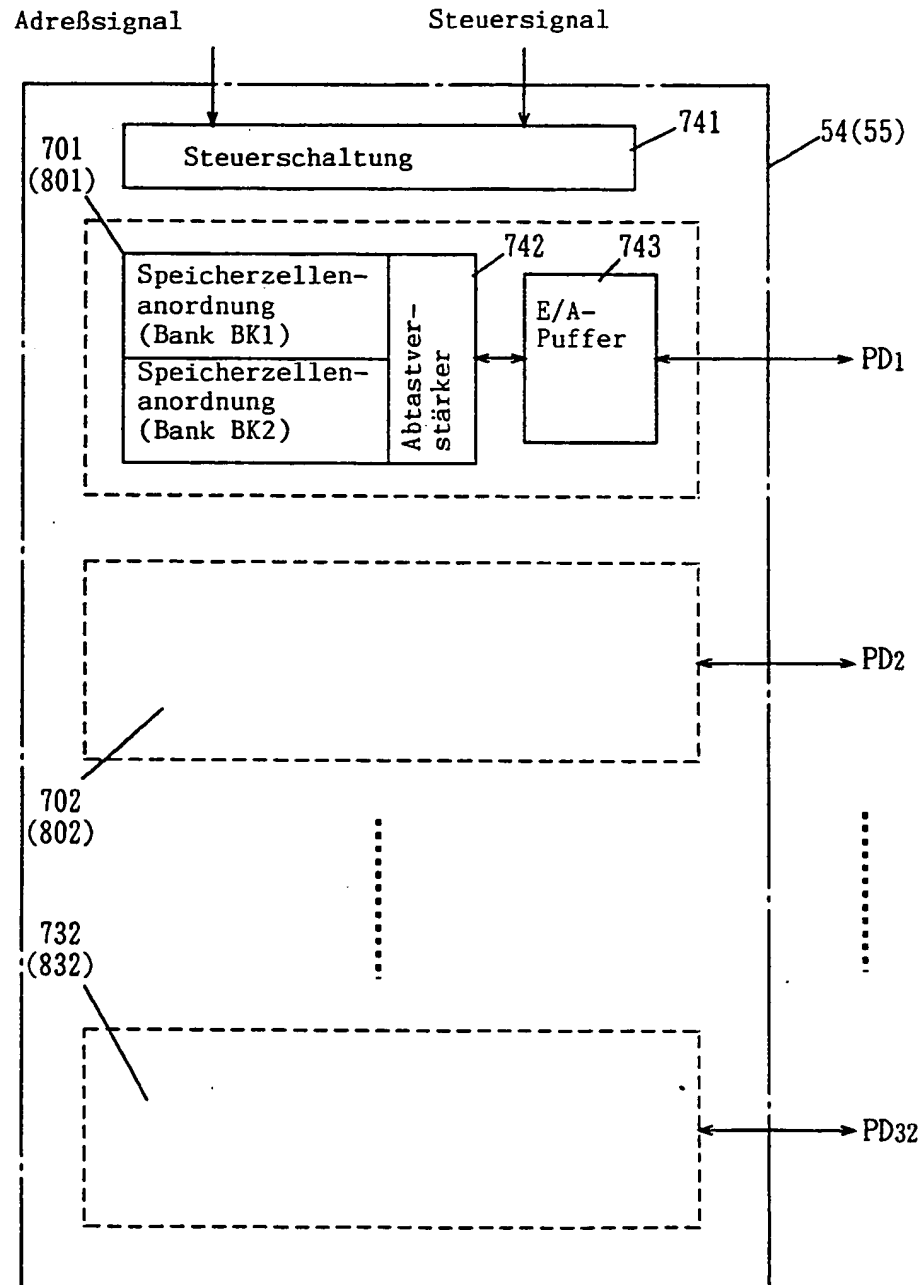


FIG. 54



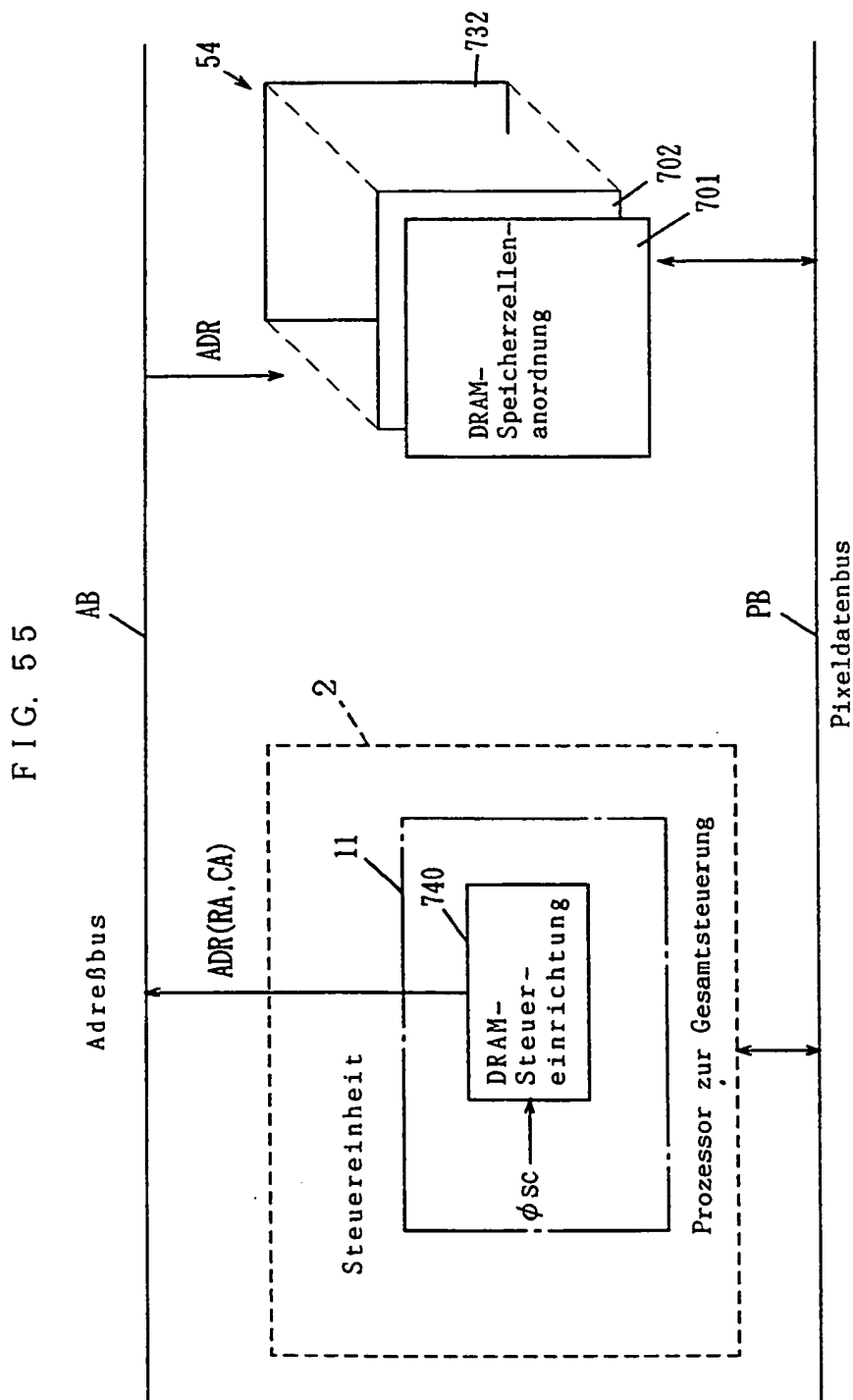


FIG. 56

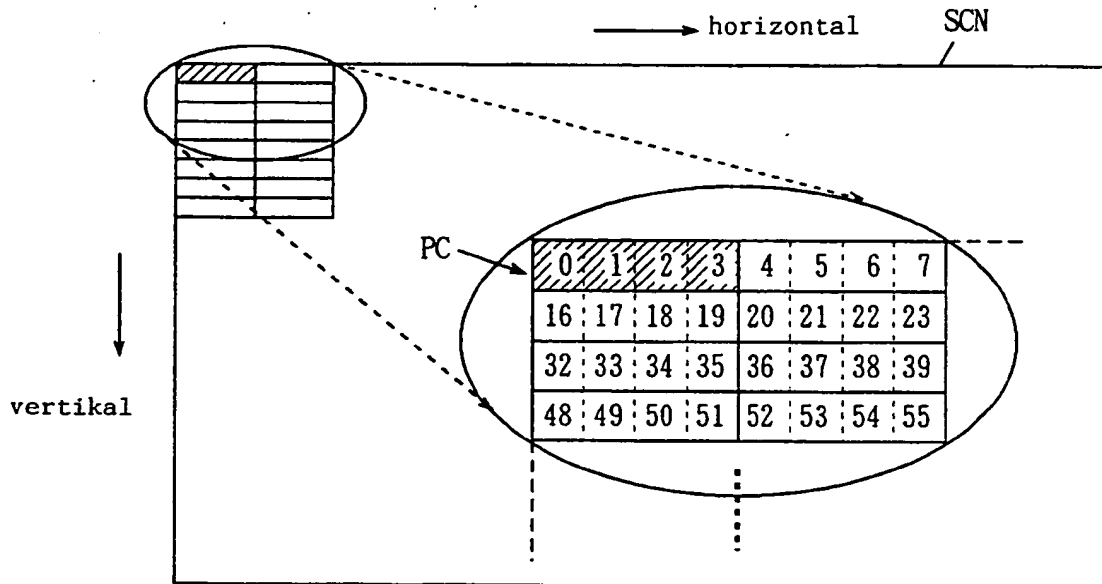


FIG. 57

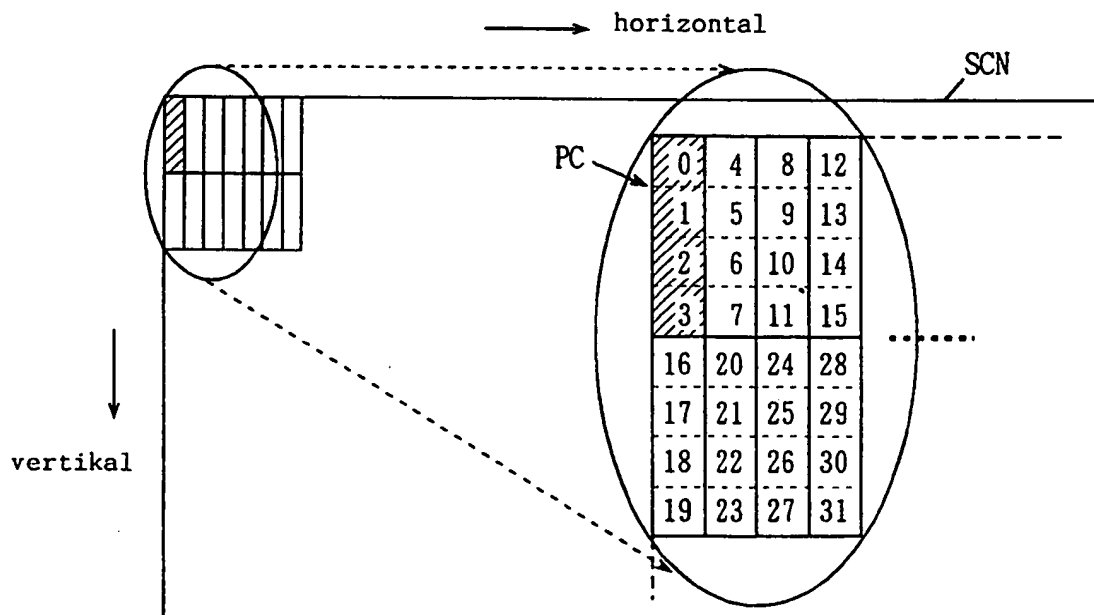


FIG. 58

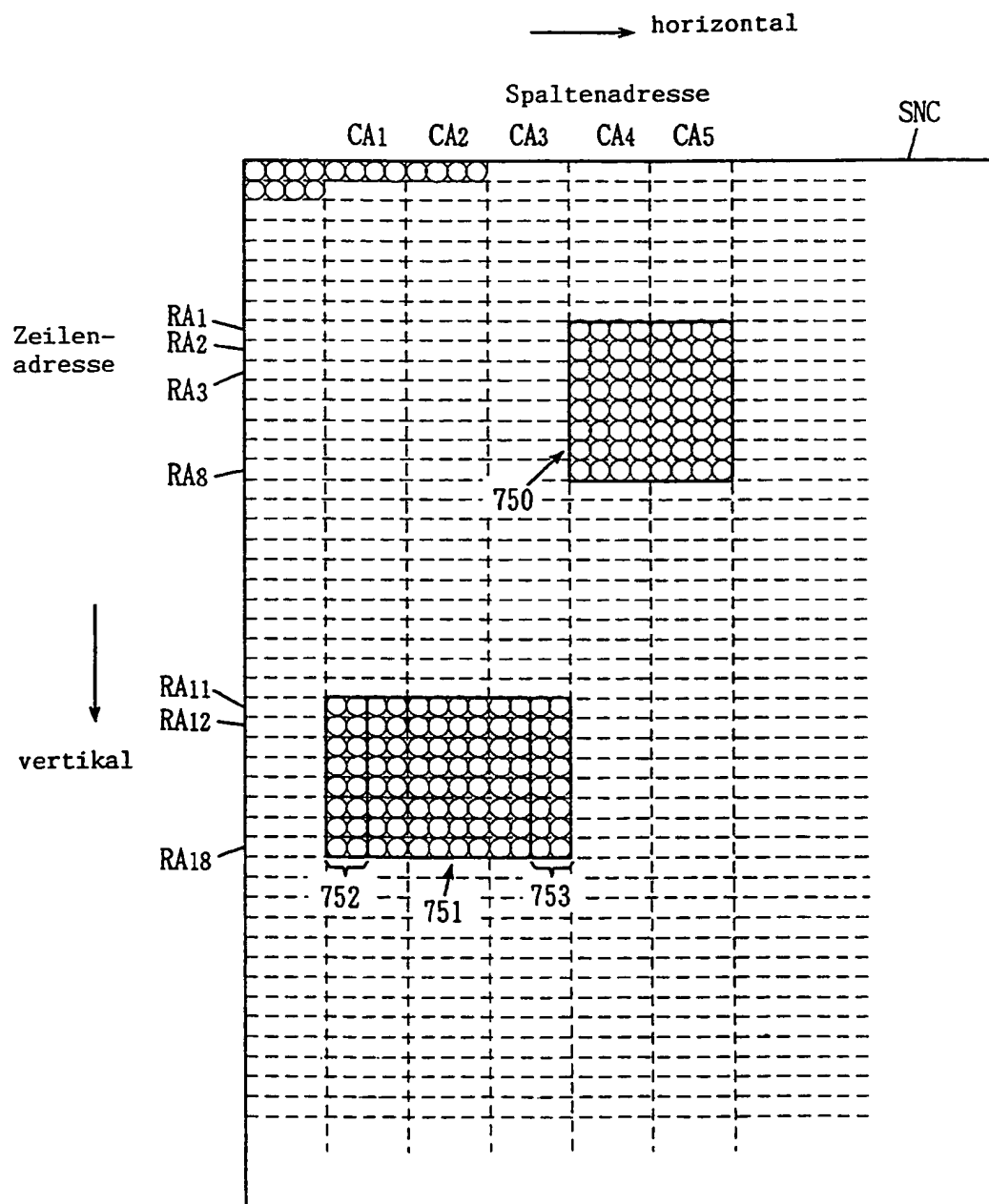
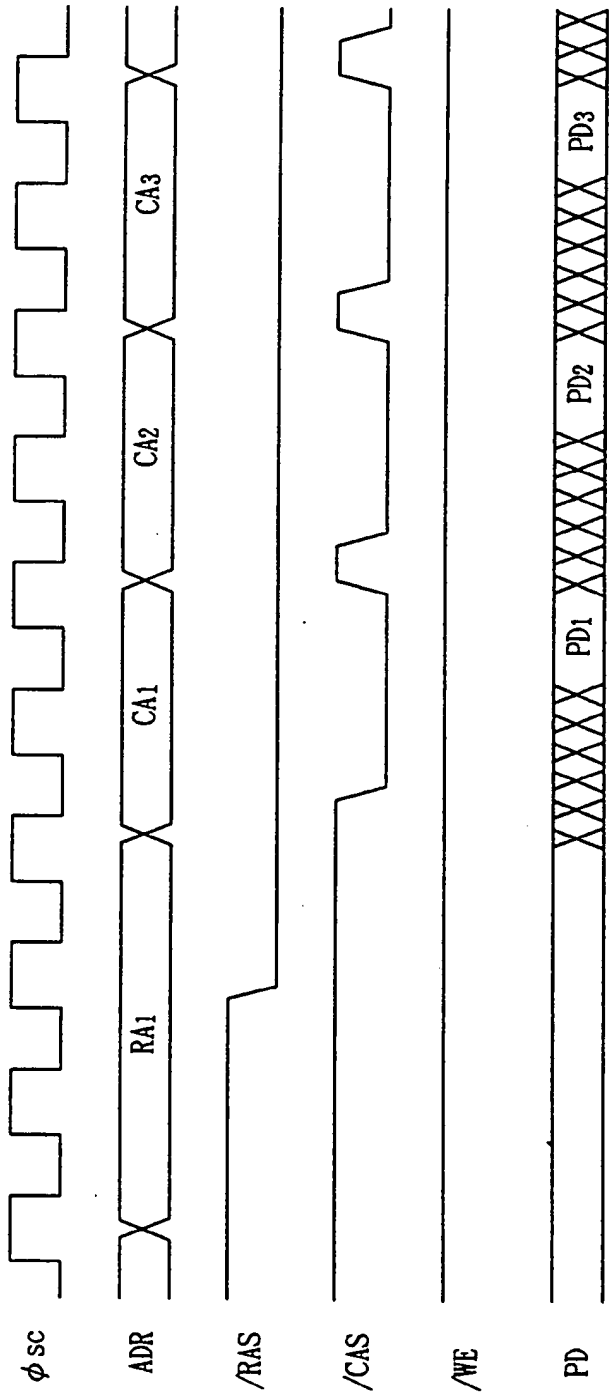


FIG. 59





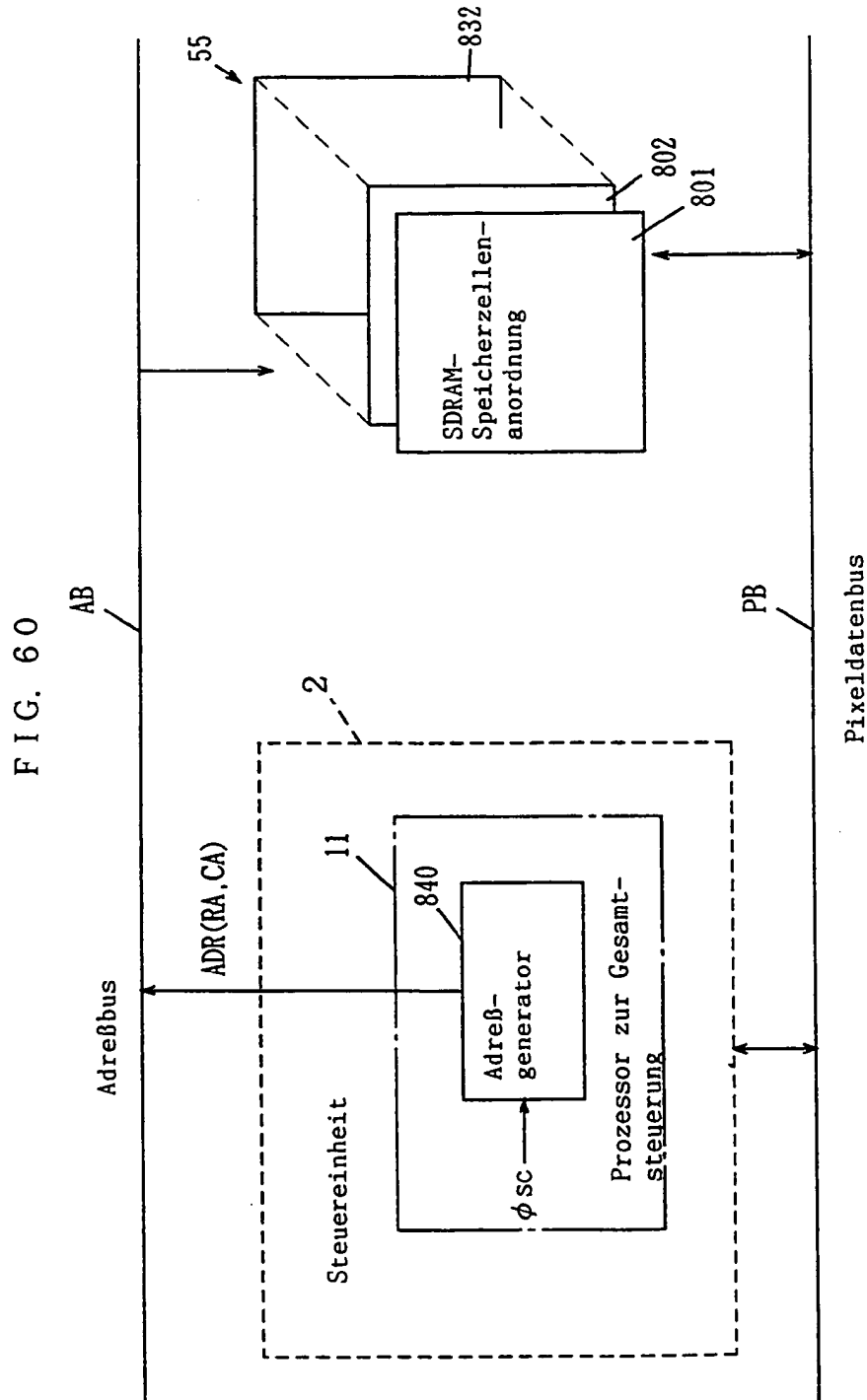


FIG. 61

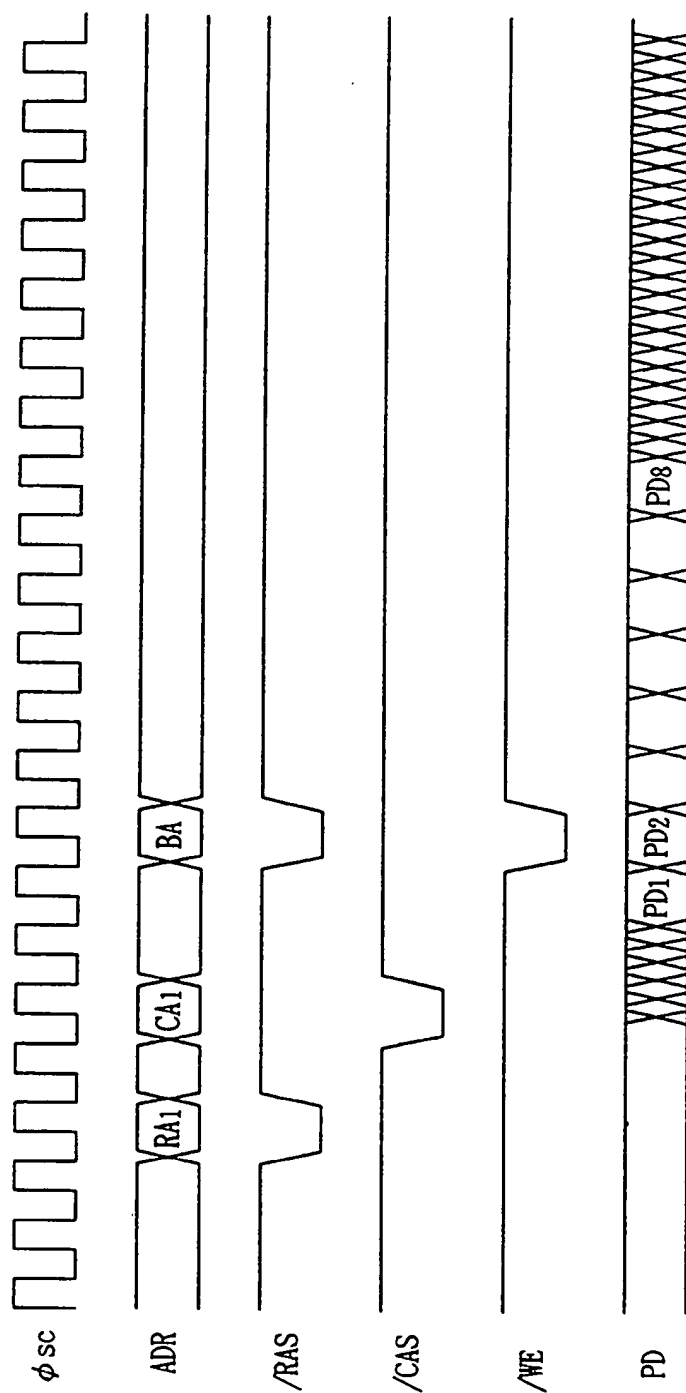


FIG. 62

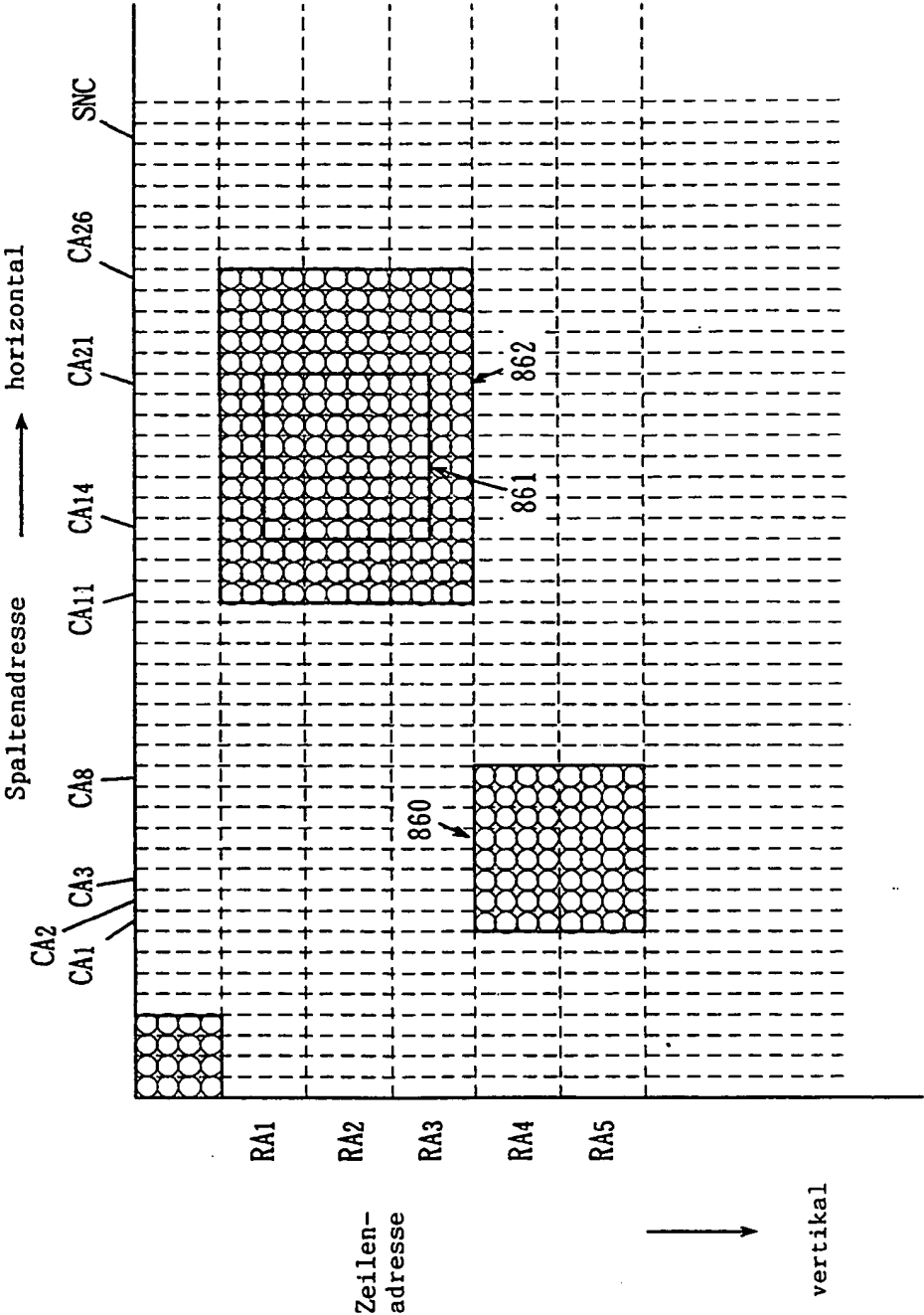


FIG. 63

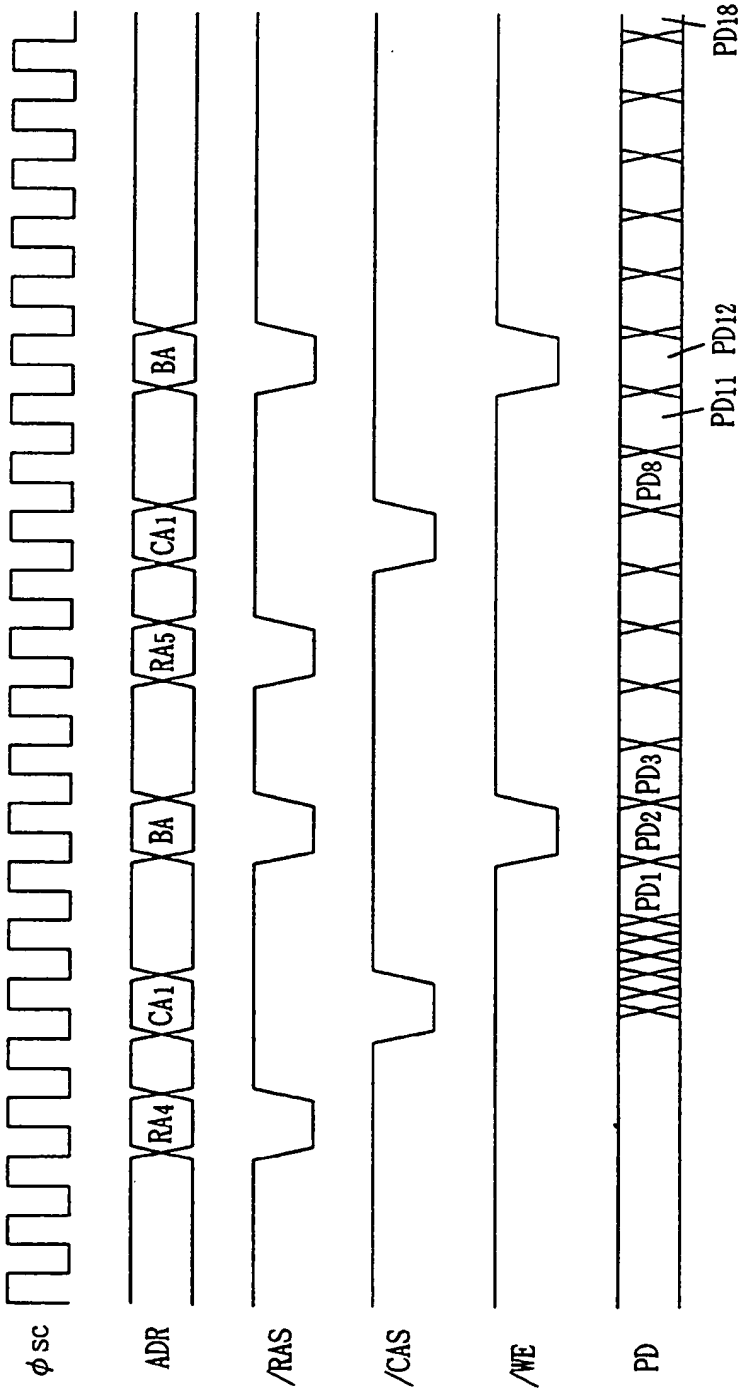


FIG. 64

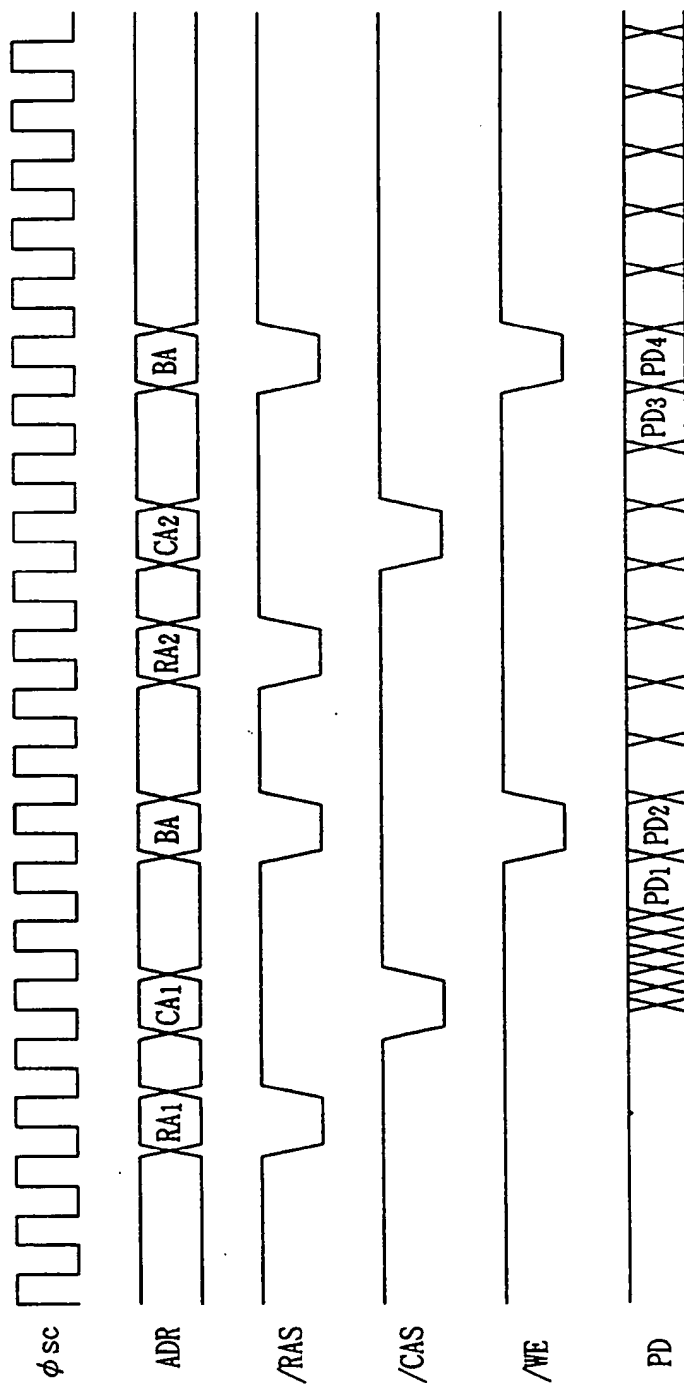


FIG. 65

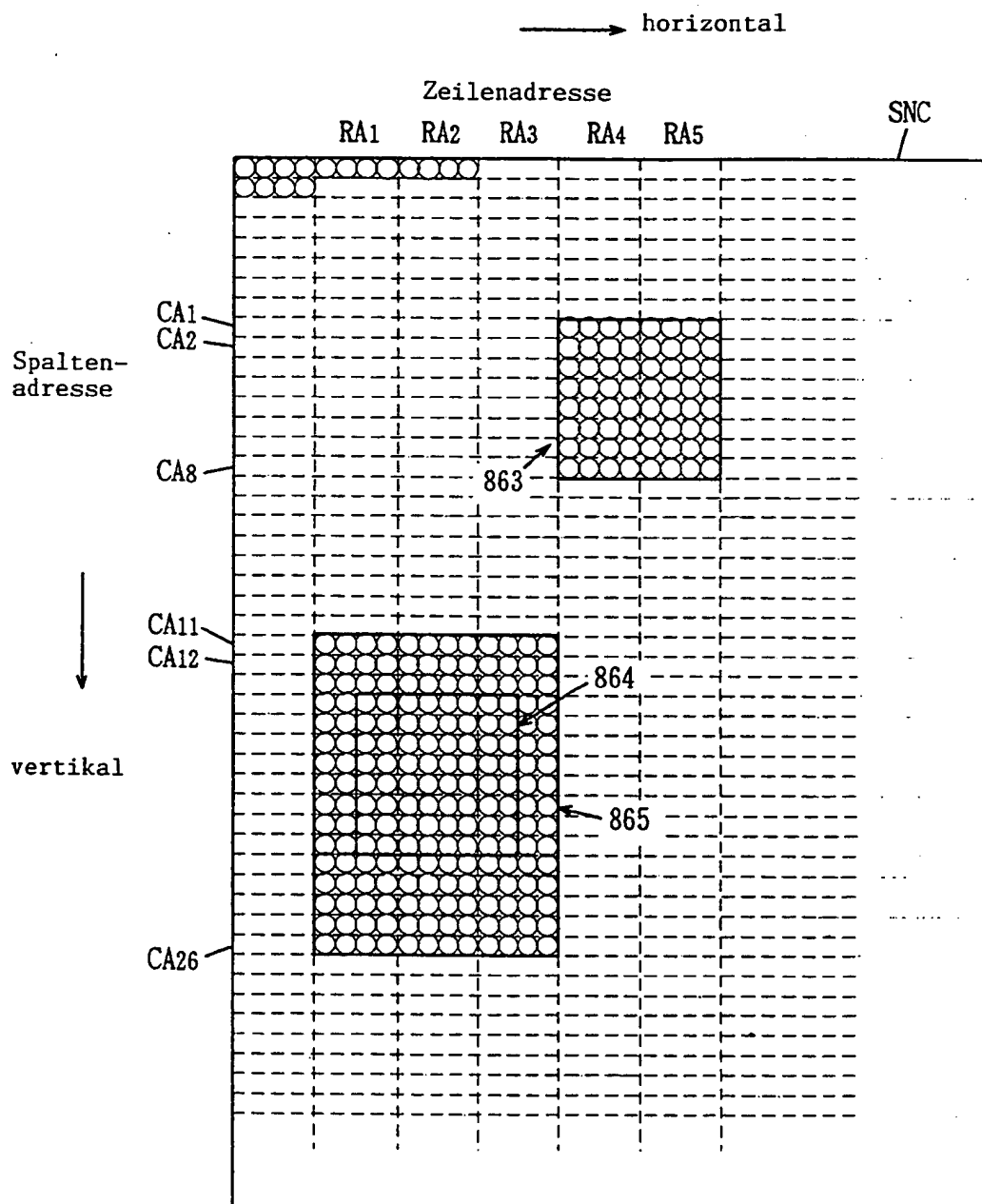


FIG. 66

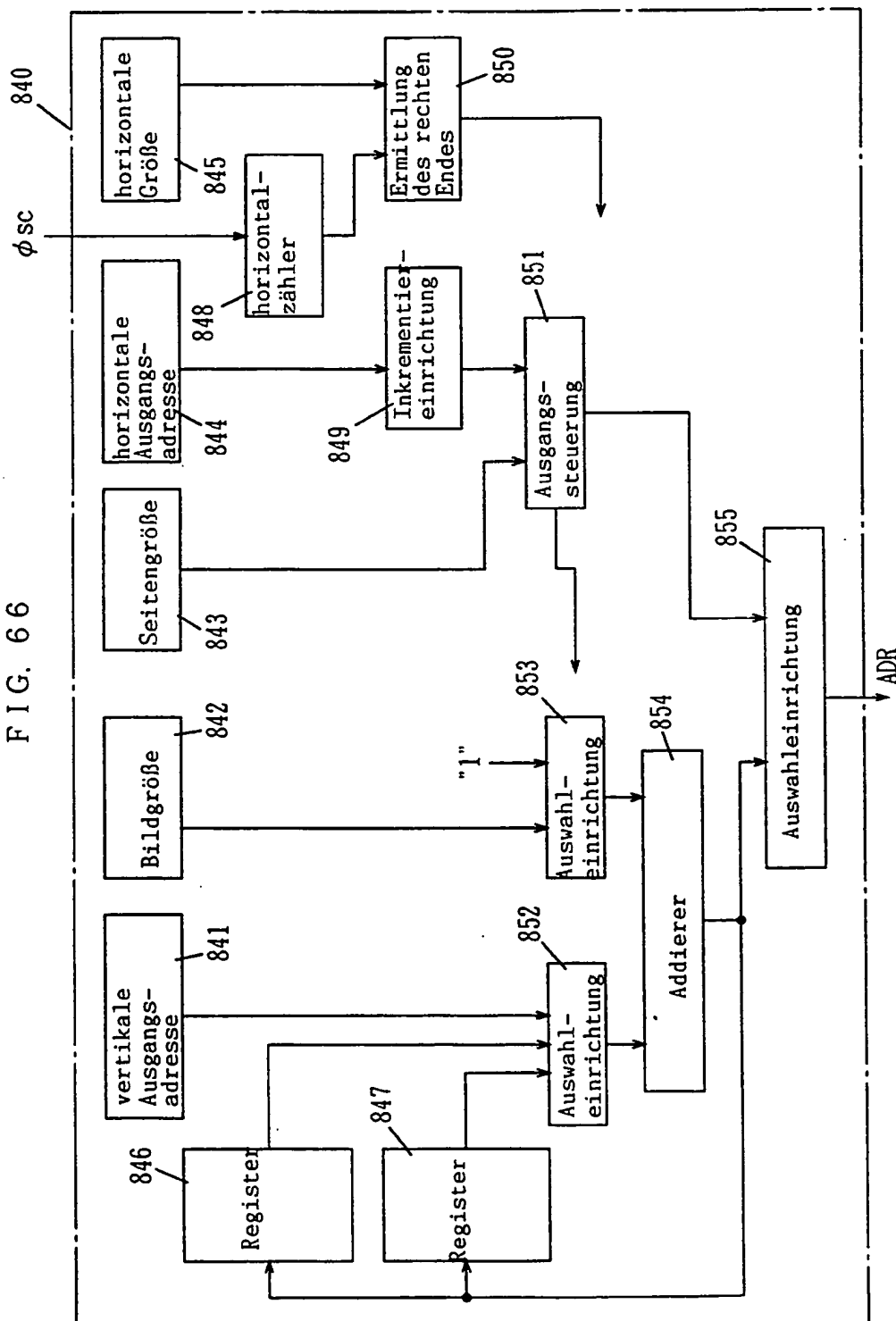


FIG. 67

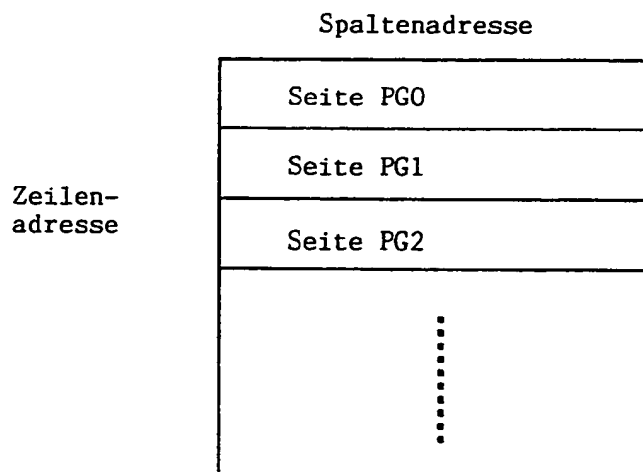


FIG. 68

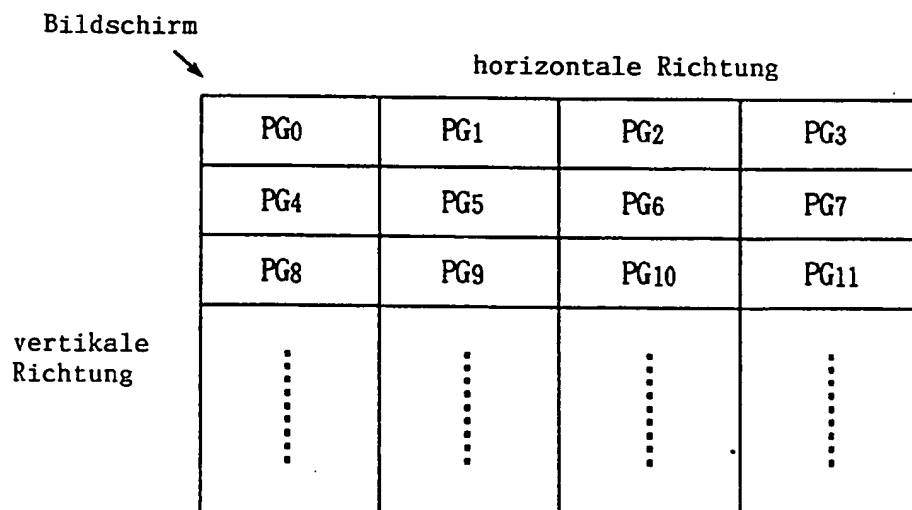




FIG. 69  
Bewegungskompensation/Bildcodierung  
Internationale Standardempfehlung zur Bildkompression

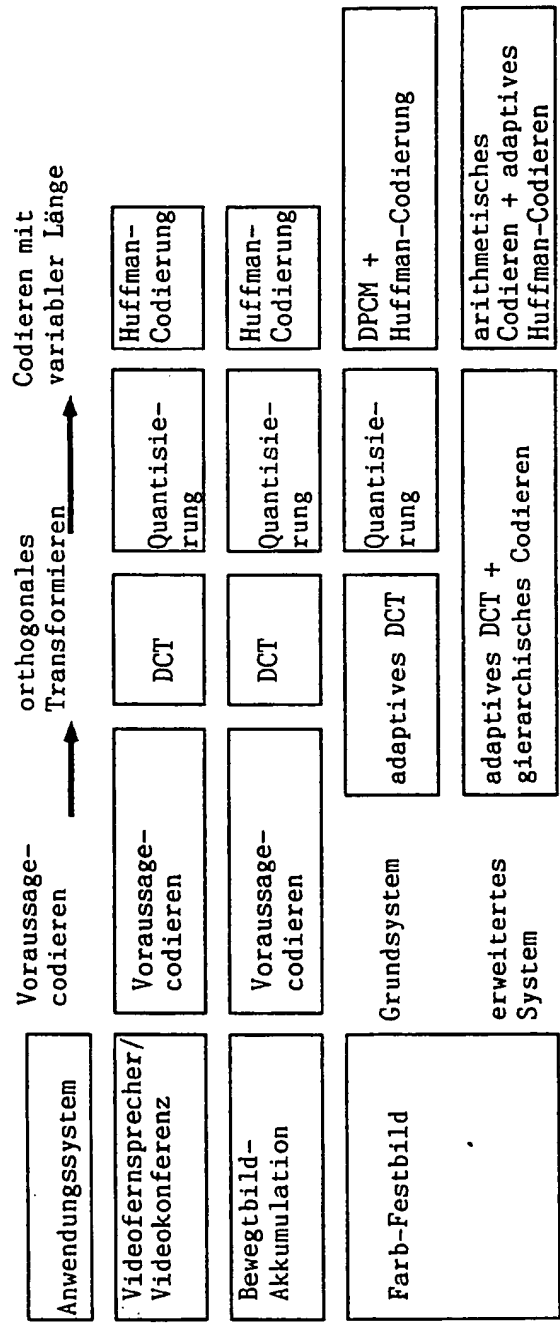


FIG. 70

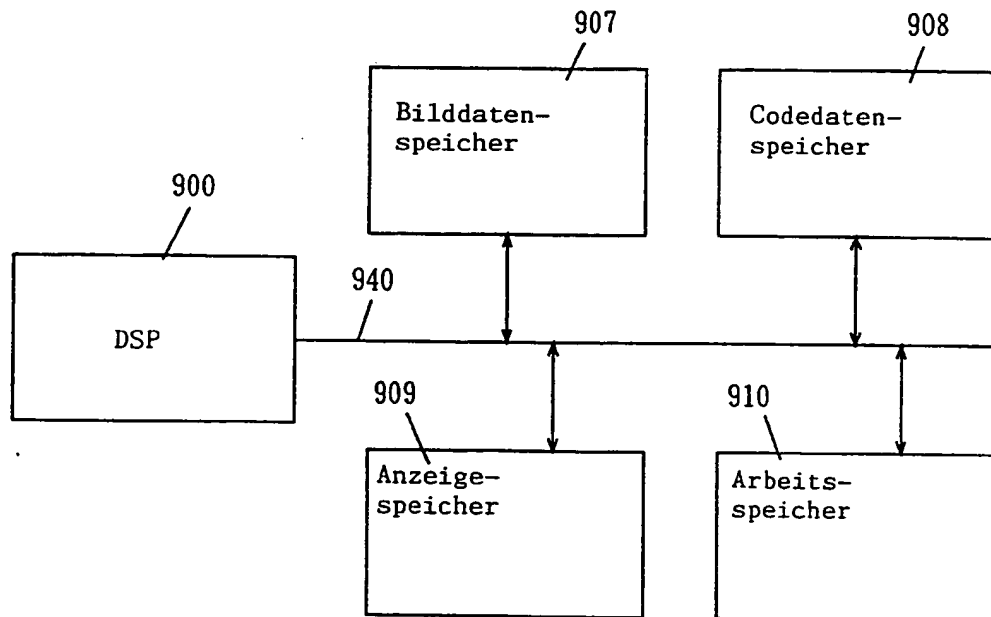


FIG. 71

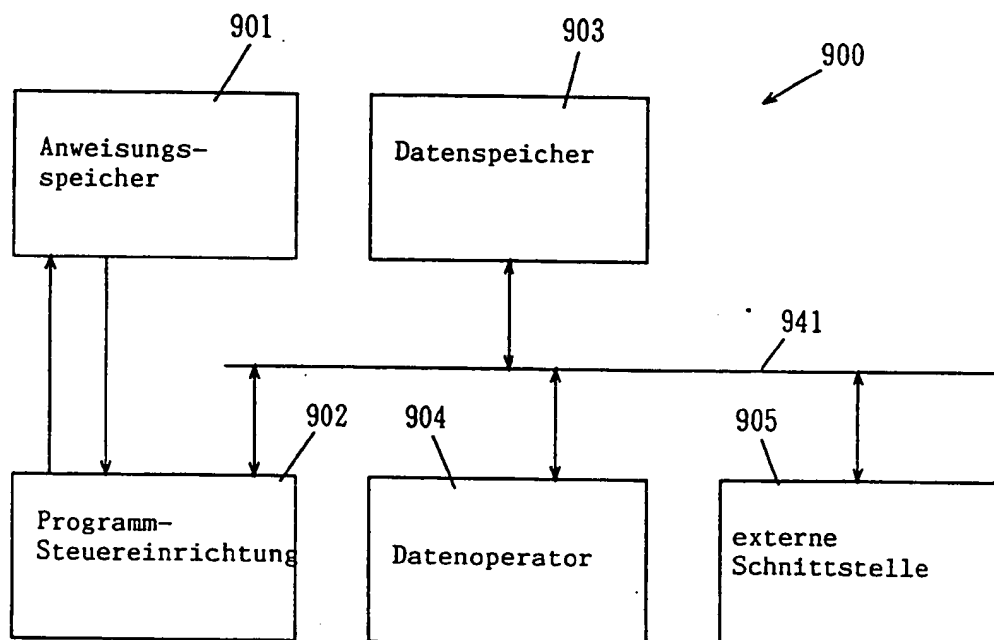


FIG.72

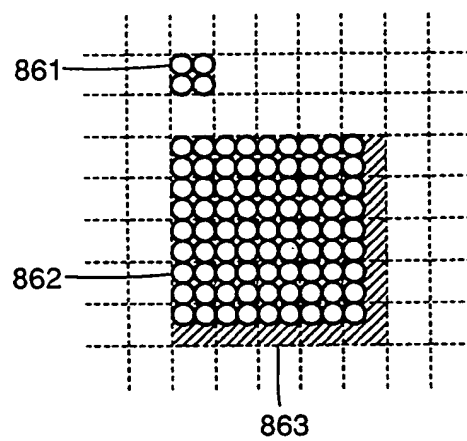


FIG.73

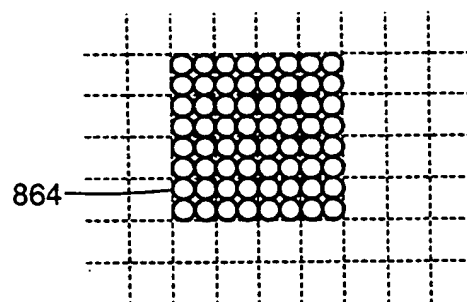


FIG.74

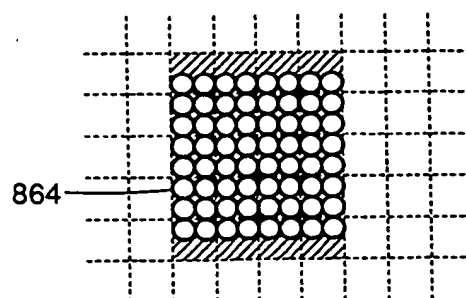


FIG.75

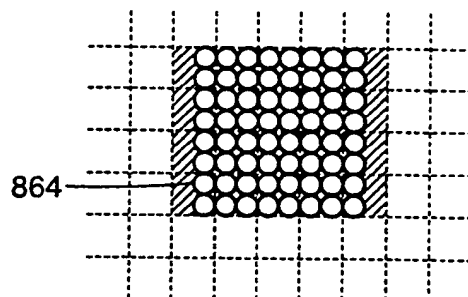


FIG.76

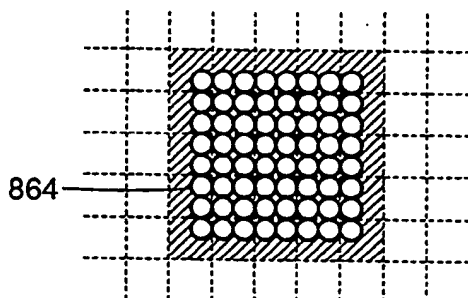


FIG.77

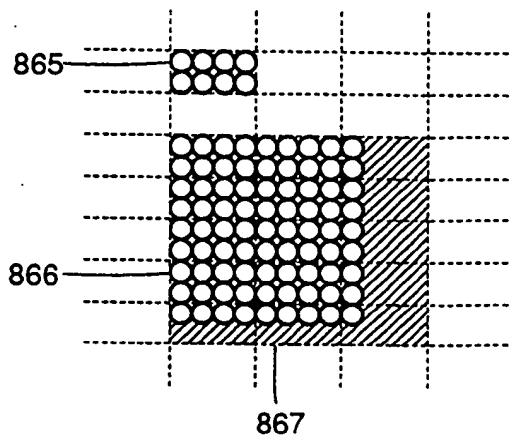


FIG.78

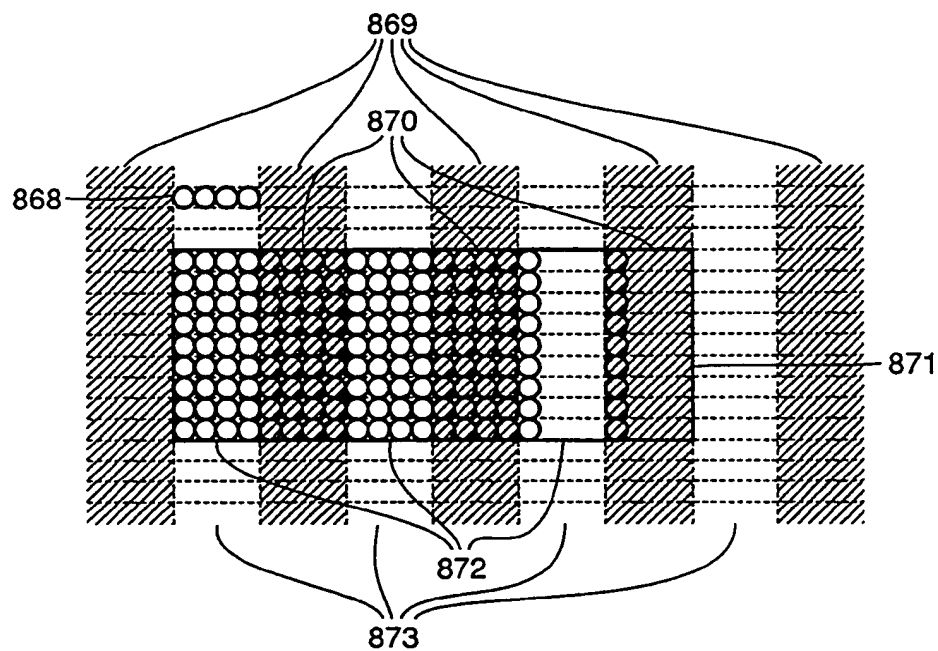


FIG.79

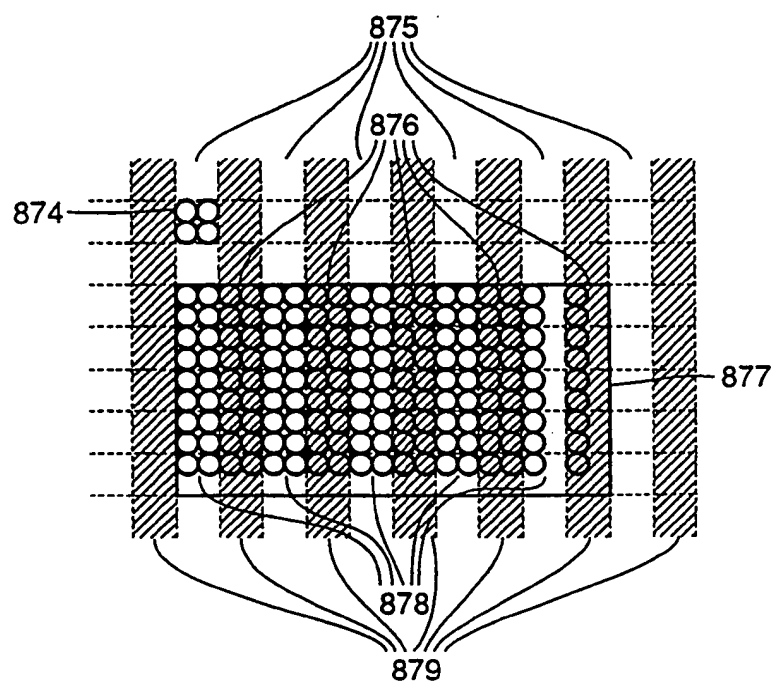


FIG.80

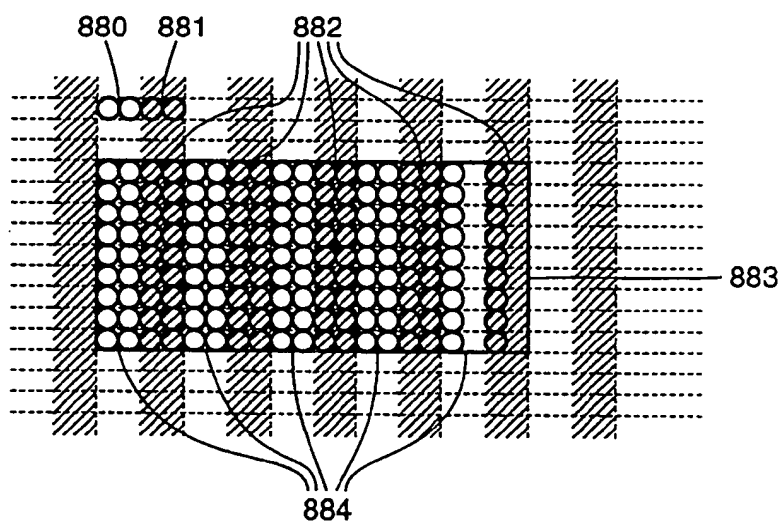


FIG.81

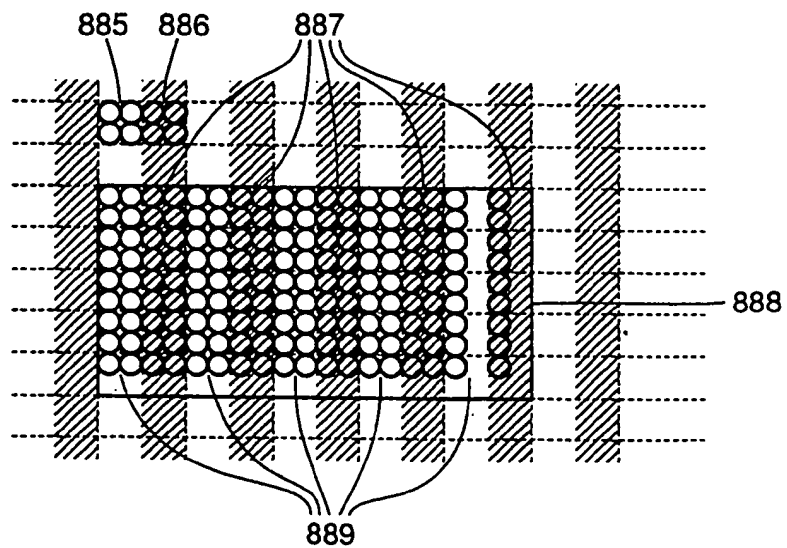
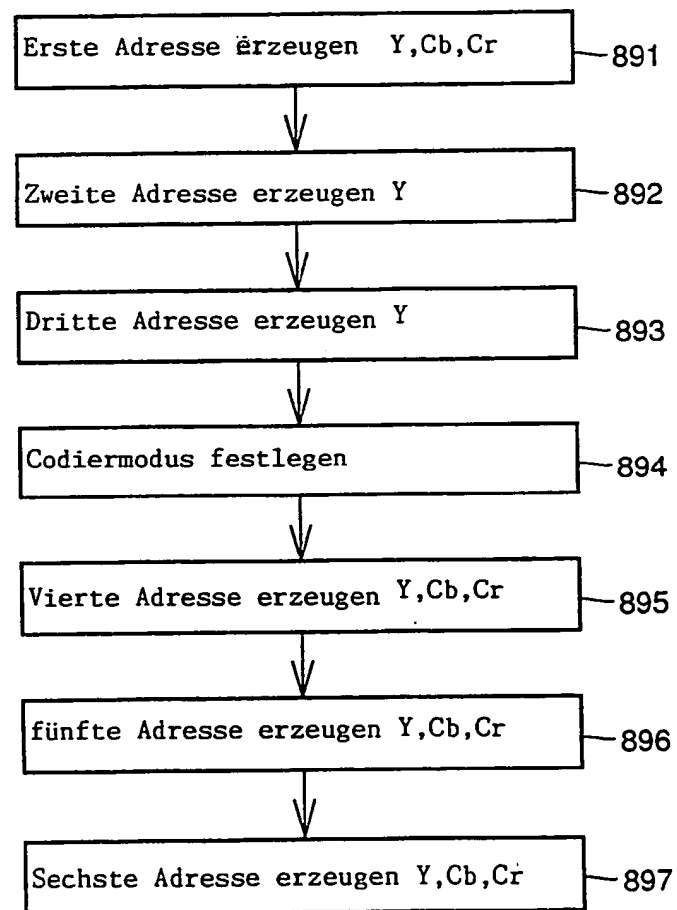


FIG.82



**FIG. 83**

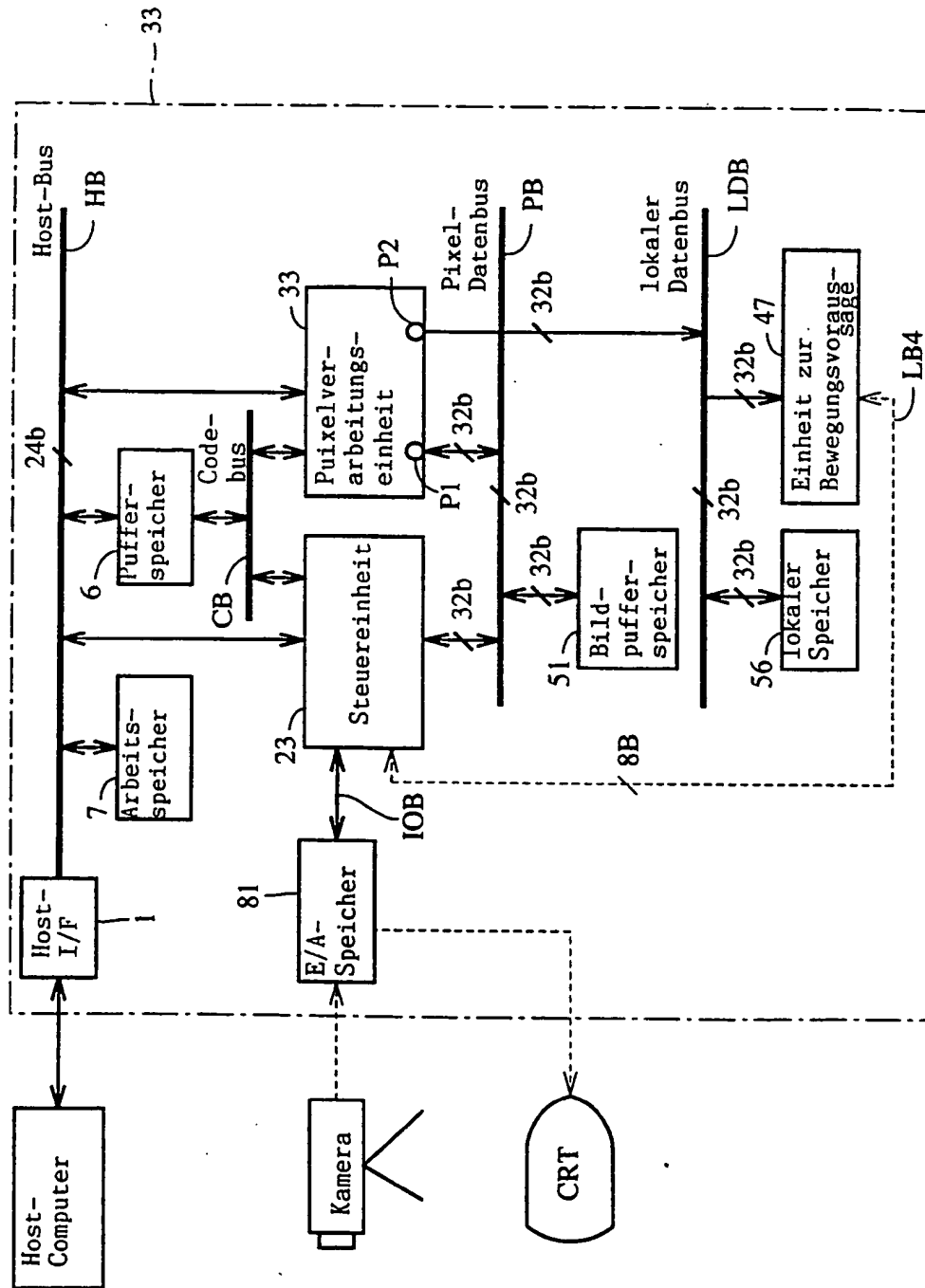




FIG.84

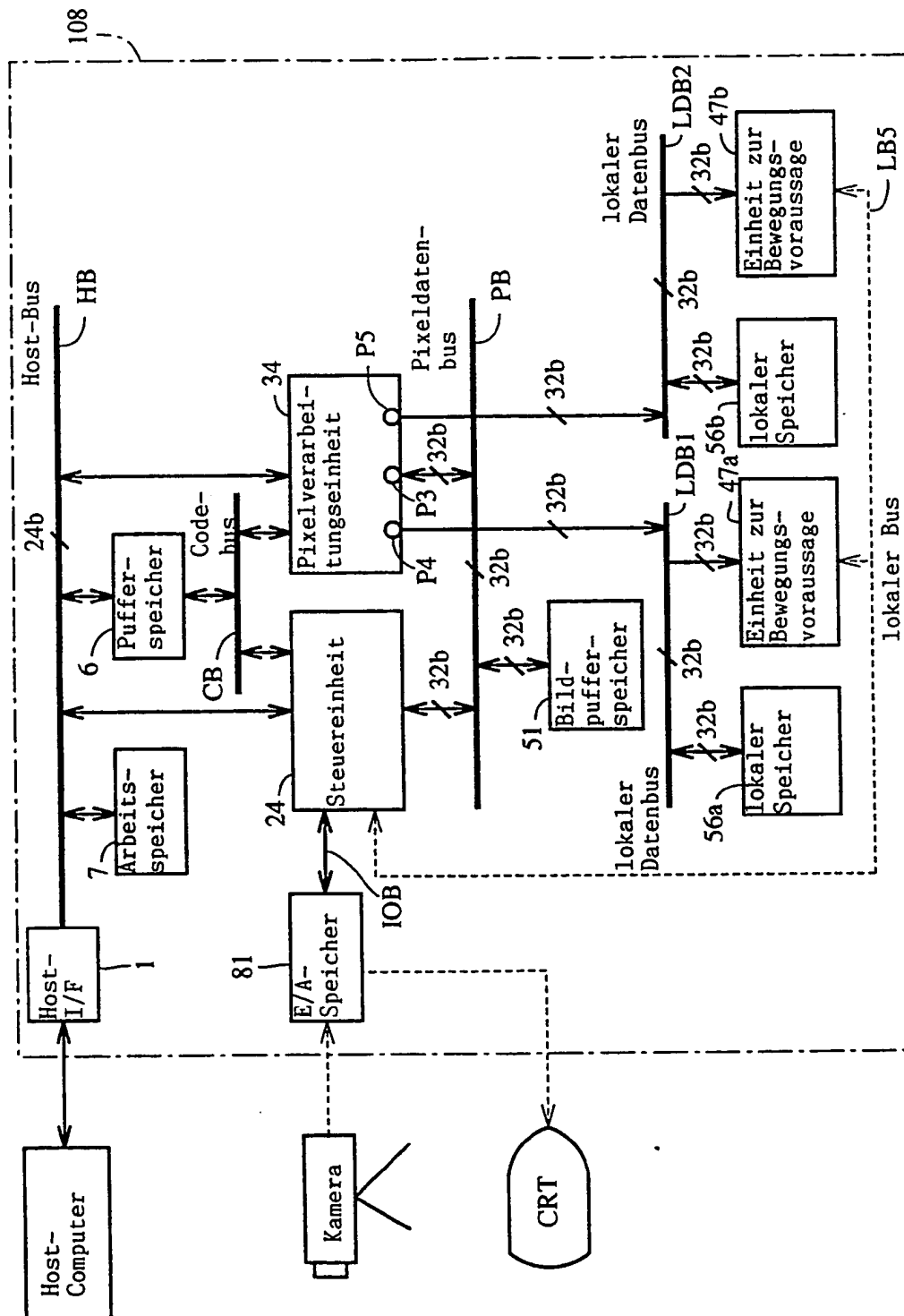


FIG. 85

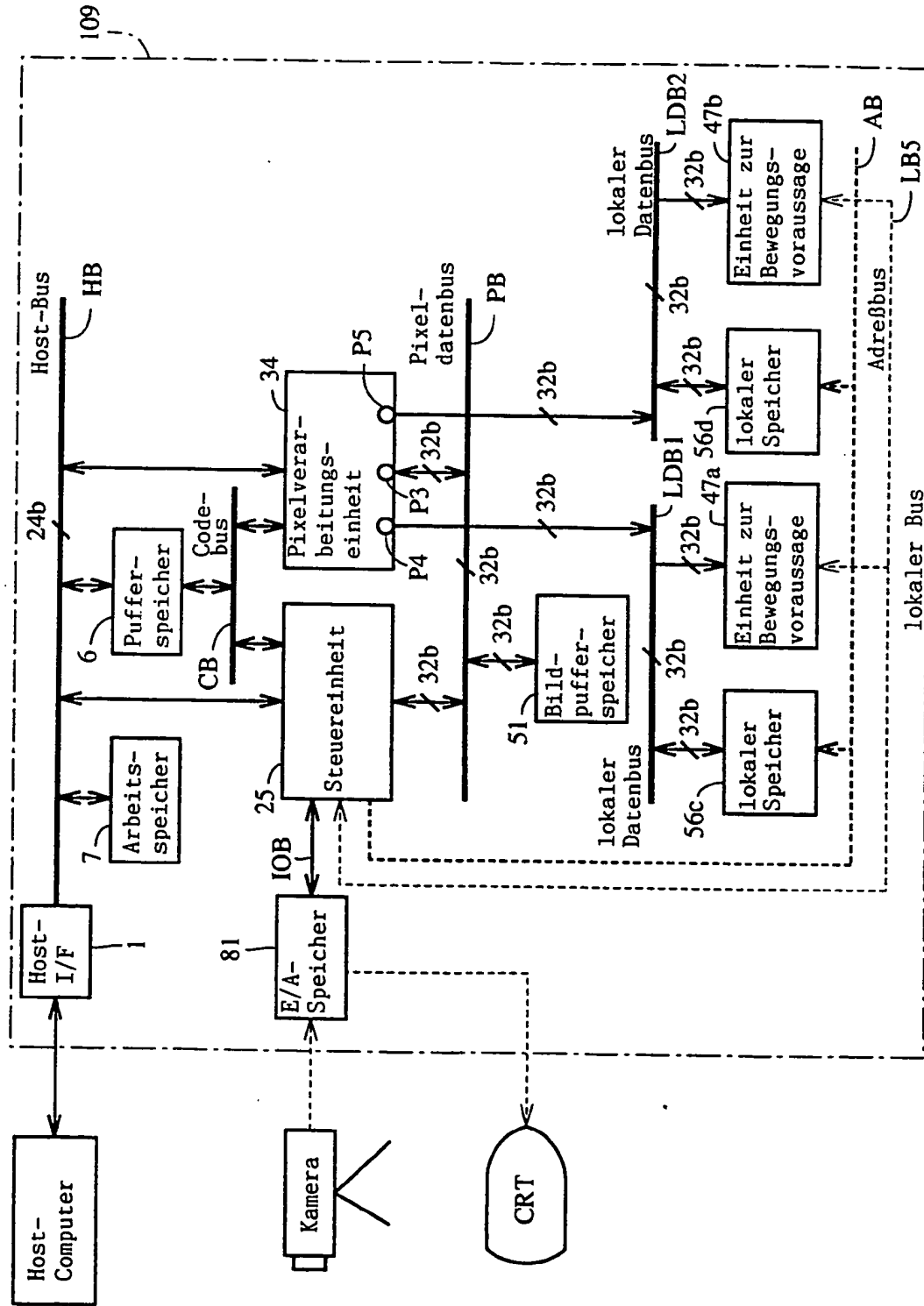
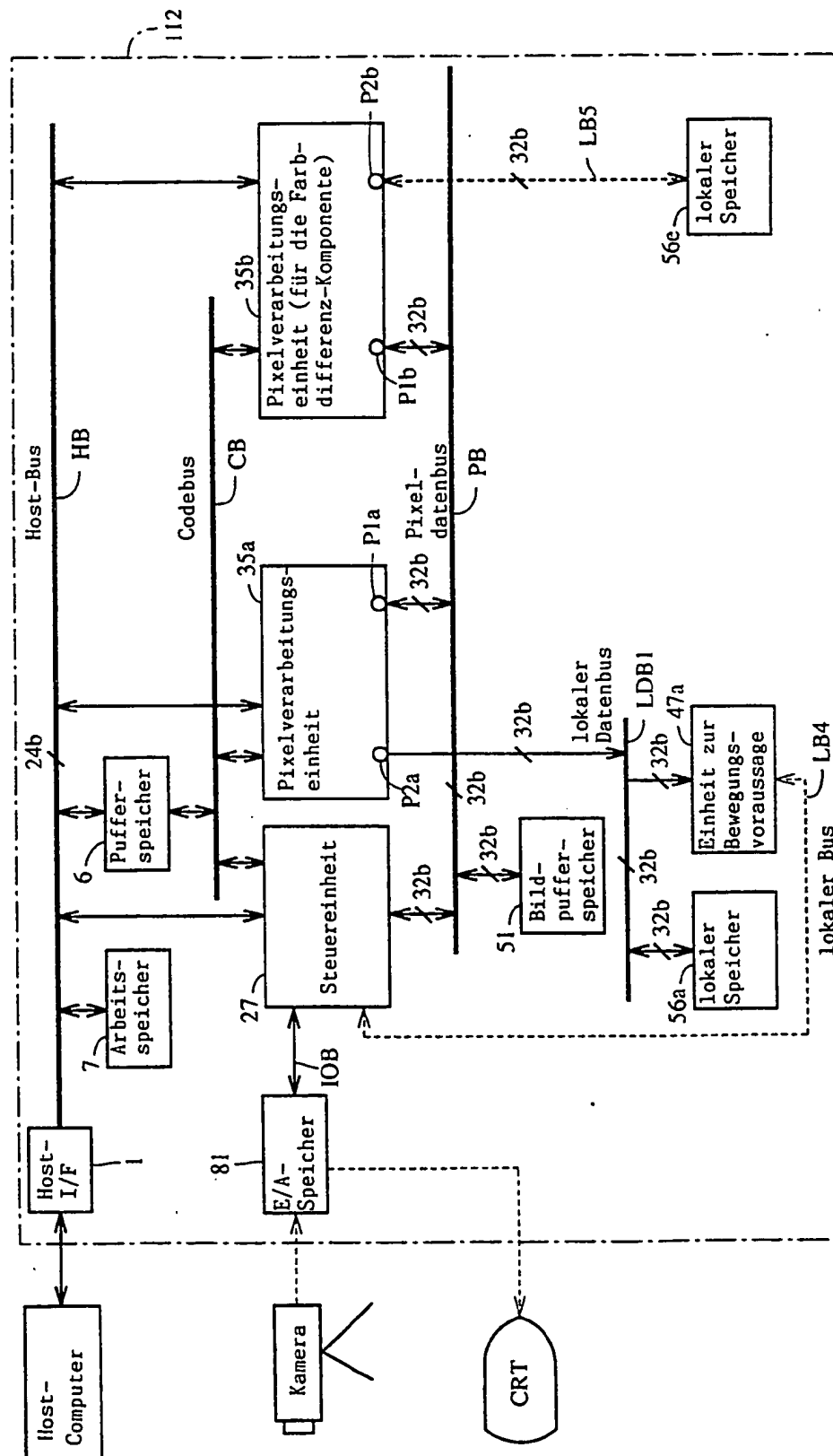






FIG. 88



**FIG. 89**

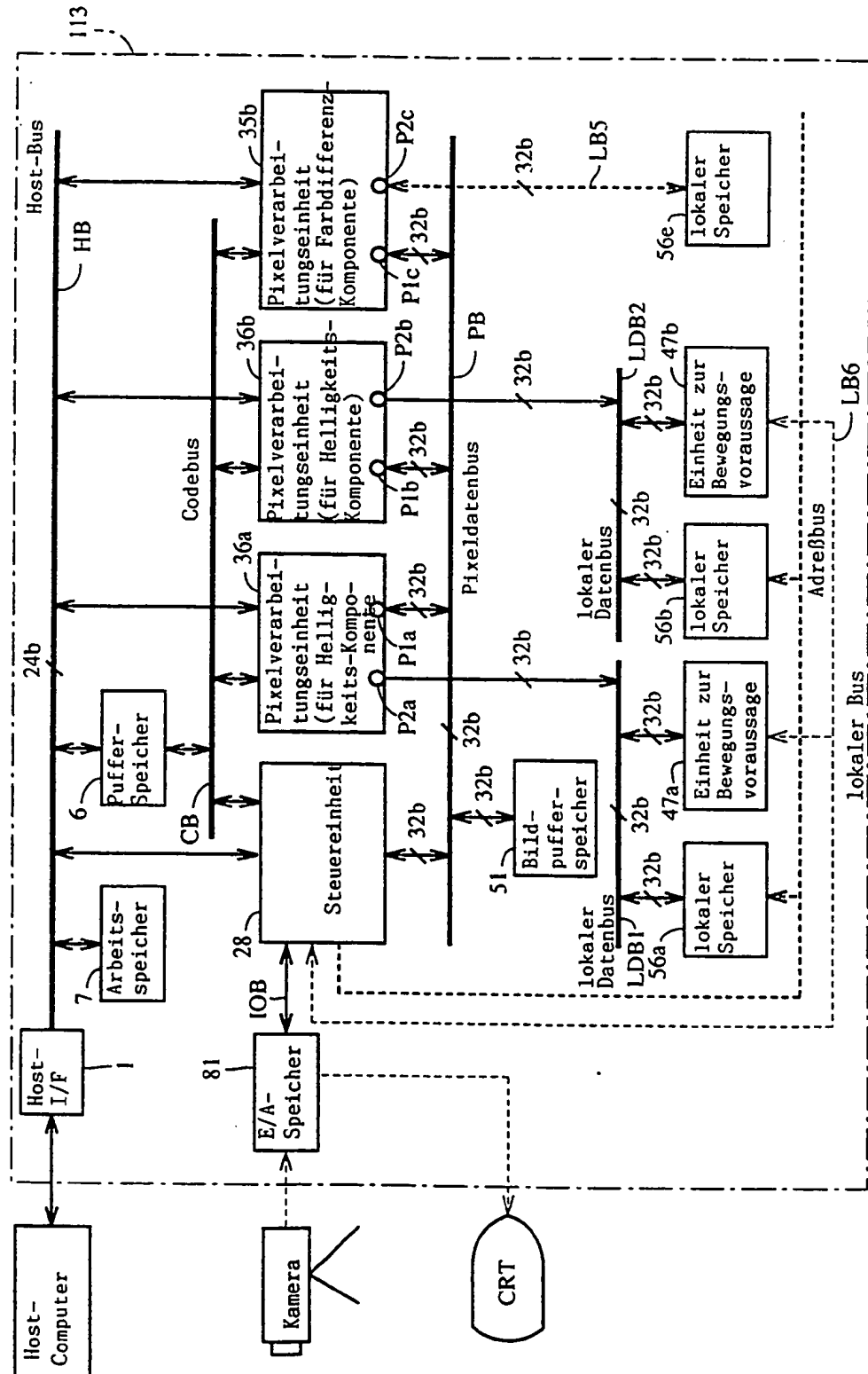
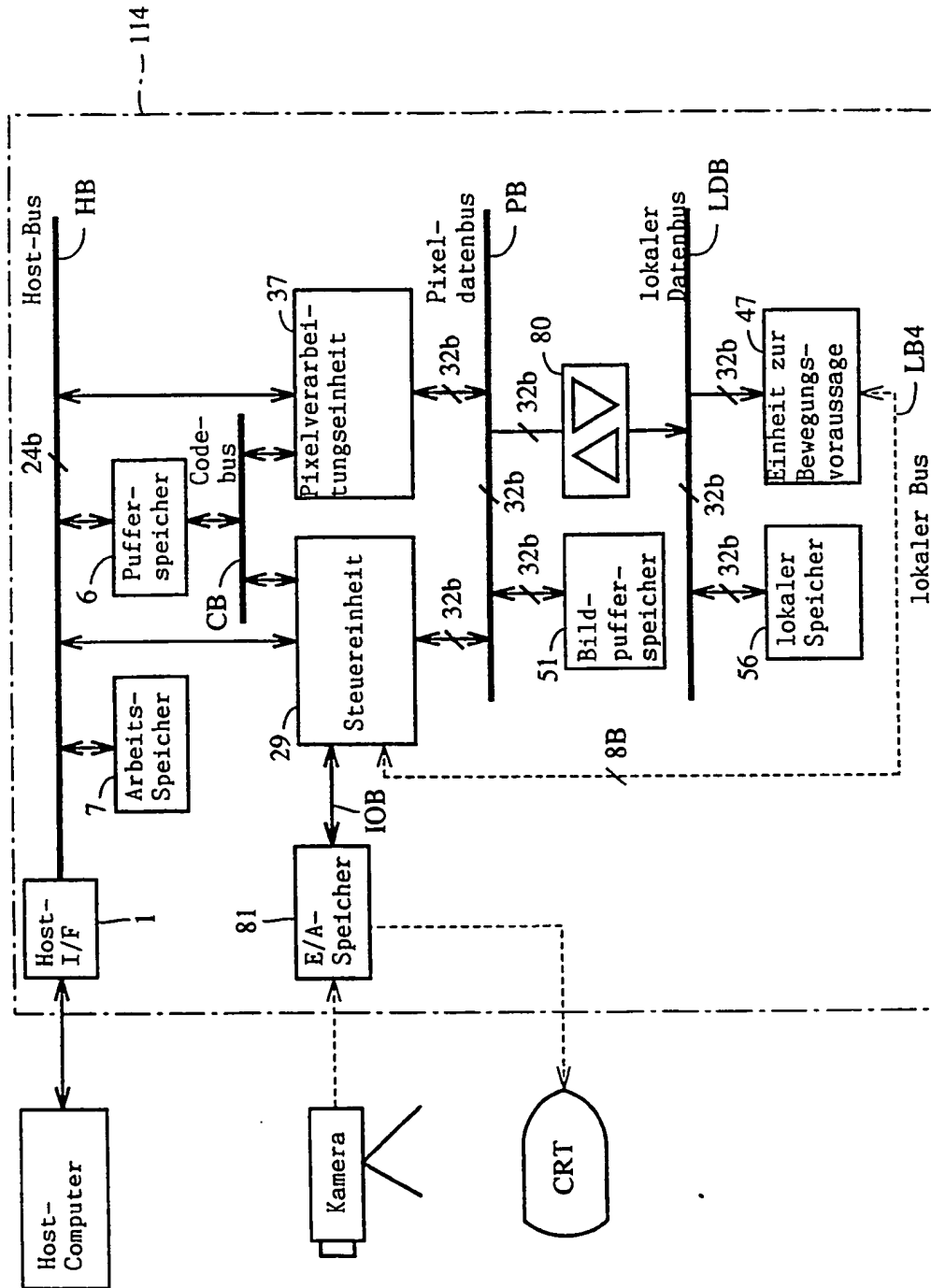


FIG.90



**FIG. 91**

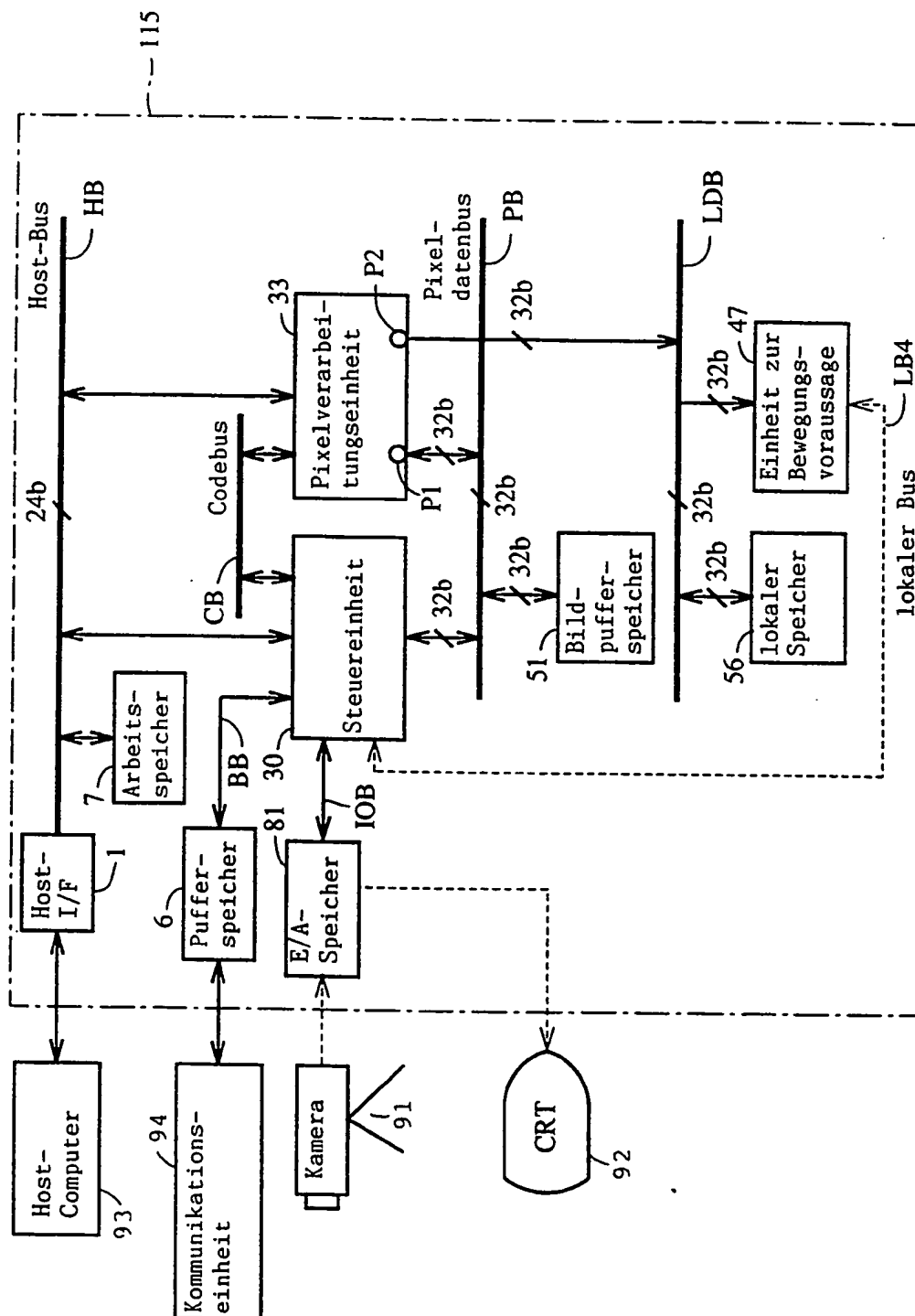




FIG.92

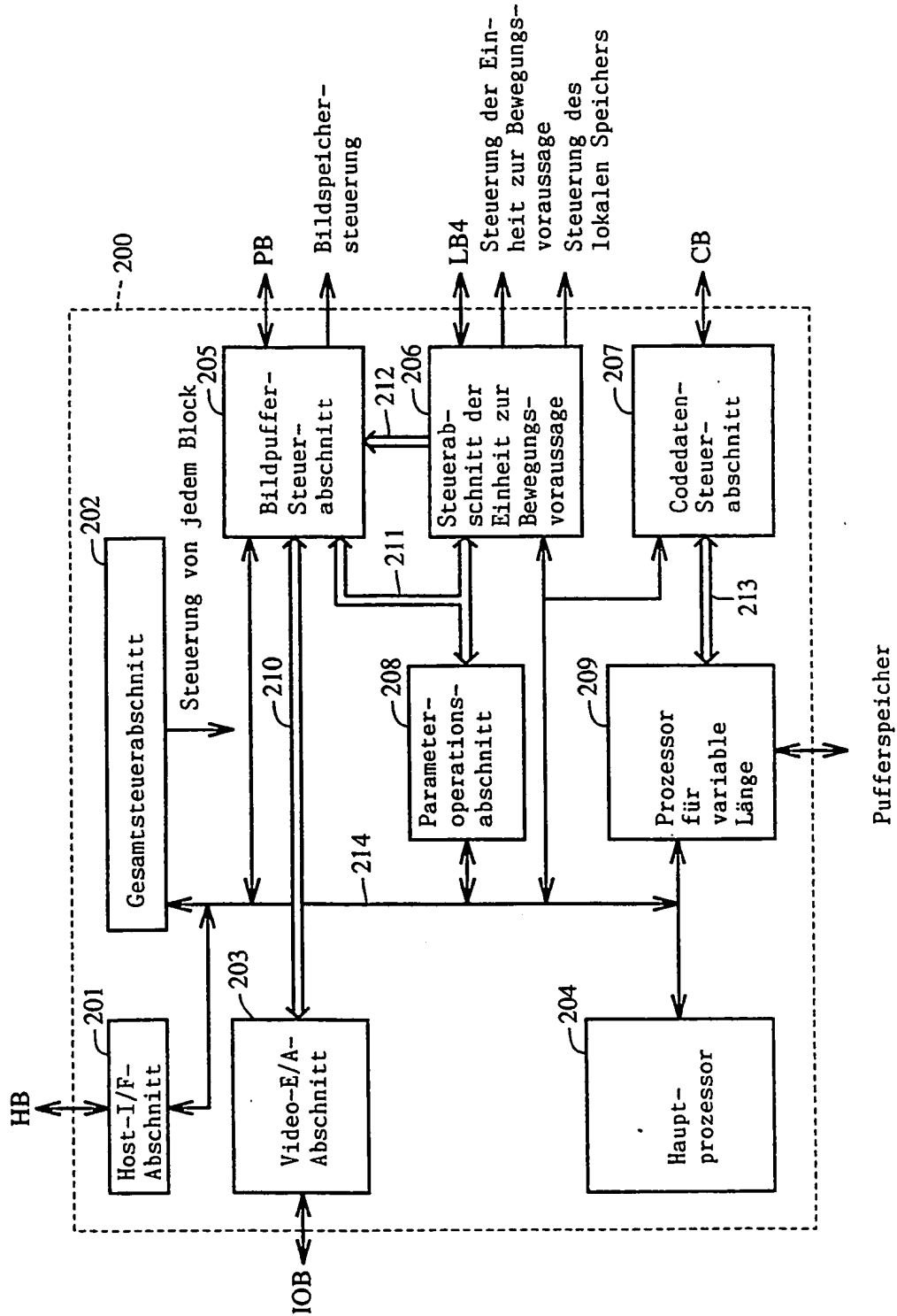


FIG.93

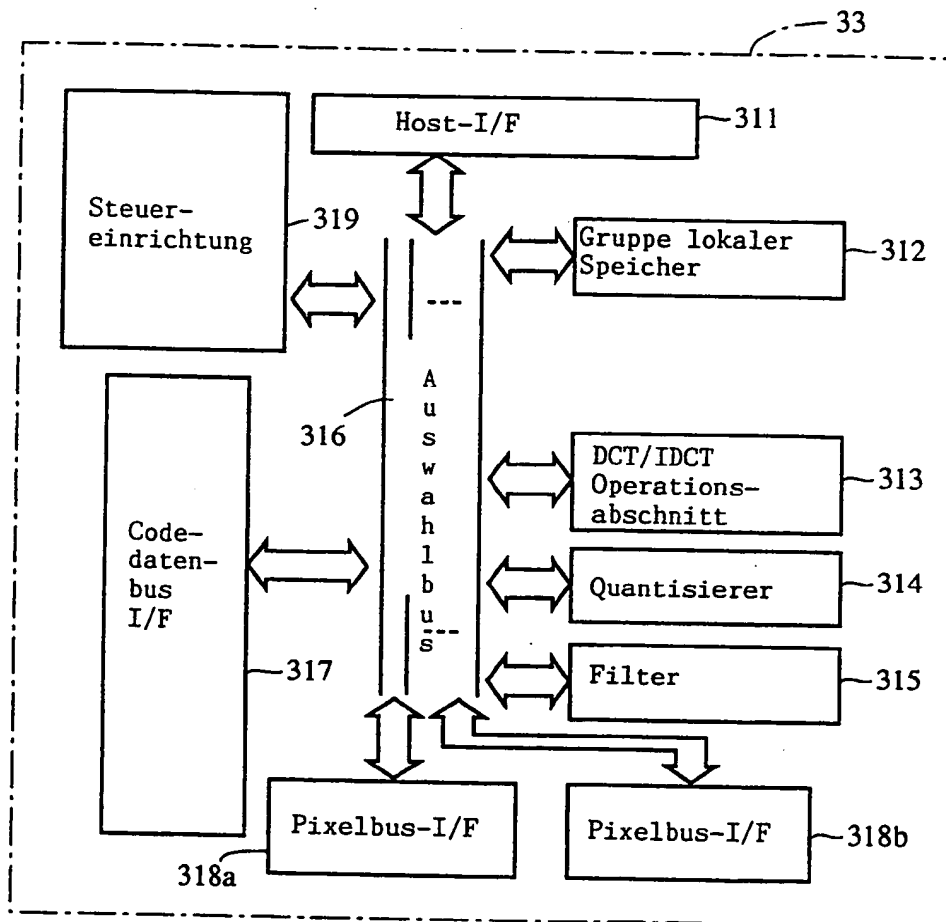


FIG.94

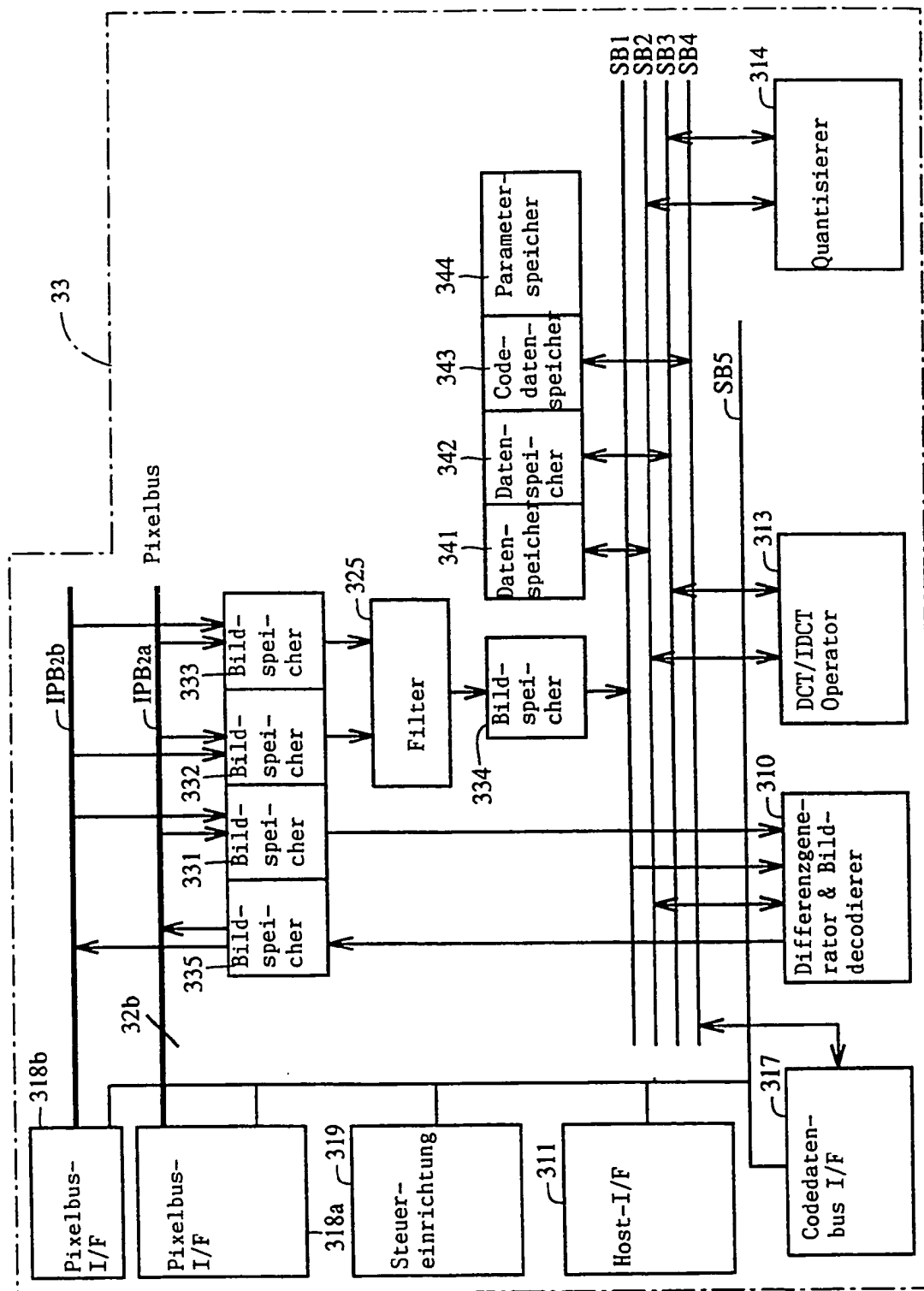


FIG.95

